INPUT

Publicación práctica para usuarios de

commodore

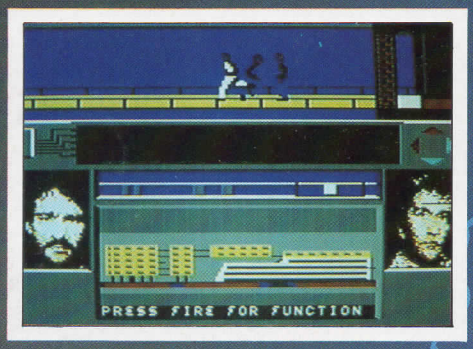
Revista mensual 1986 Precio 350 Ptas Año 1 Número 11

FABULOSO AMIGA

MODELIZANDO LA REALIDAD

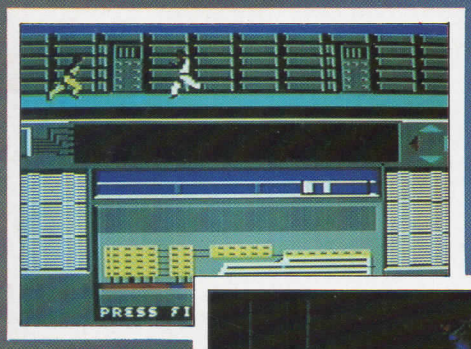
MUSICA MIENTRAS TRABAJAS

Commodore 64/128
Amstrad 464/664/6128
Spectrum 48/128



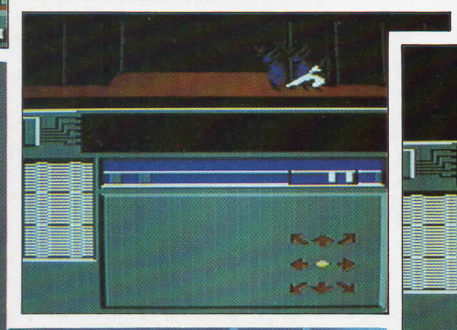
Tienes una misión: Destrozar la banda de traficantes que ha secuestrado a tu amigo. Puedes dar patadas, puñetazos, y saltos, pero recuerda que los jefes del tráfico de drogas son muy poderosos. Solo N.E.X.U.S. puede ayudarte a salir victorioso.

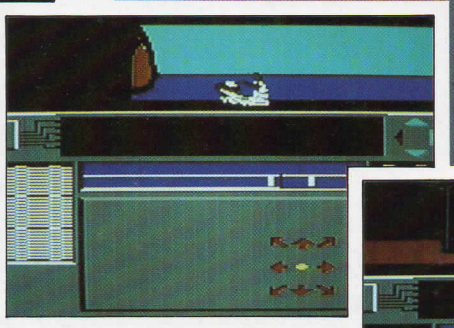
NEXUS

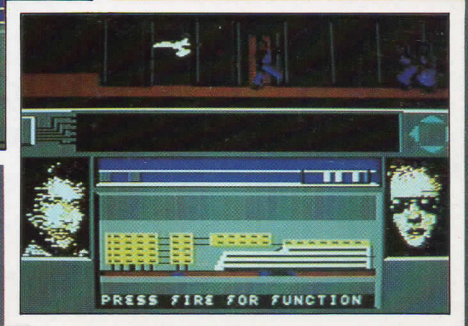


Una aventura de arcade con mucha acción y llena de peligros.

Es necesario joystick
Aprende jugando
Pide una demostracion ahora mismo.







U.S. GOLD
All American Software

U.S. Gold (S.A.),
Santa Engracia 17,
Sexto,
28010 Madrid.

AÑO 1 NUMERO 11

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Luis R. Palencia, Christophe Païs, Francisco Tórtola, Benito Román, Esther de la Cal.
PORTADA:
Foto, Nieves; Maquillaje y peluquería, Grupo Cinco; Modelo, Susana Alguacil.
INPUT Commodore es una publicación juvenil de EDICIONES FORUM
GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ General Varela, 35
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99
FOTOMECANICA: Ochoa, S. A.
COMPOSICION: EFCA, S. A.
IMPRESION: Sirven Grafic
C/ Gran Vía, 754-756. 08013 Barcelona
Depósito legal: M. 27.884-1985
SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12
REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Teléf. 250 10 00
DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona.

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT Commodore es independiente y no está vinculada a Commodore Business Machines o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío.

Copyright ilustraciones del fondo gráfico de Marshall Cavendish, págs. 22, 23, 24, 26, 27, 31, 32, 33, 34, 36, 37, 40, 41, 42, 43, 44, 47.

INPUT commodore

SUMARIO

EDITORIAL 4

ACTUALIDAD 6

REVISTA DE HARDWARE
FABULOSO AMIGA 8
SOUND SAMPLER, MUSICA AL BYTE 16
QUIEN ASESINO A LOS JUEGOS 30

PROGRAMACION
PONIENDO LAS COSAS EN ORDEN 22
COMO SE ALMACENAN LOS PROGRAMAS 50

APLICACIONES
MODELIZANDO LA REALIDAD 40

CODIGO MAQUINA
MUSICA MIENTRAS TRABAJAS 46

REVISTA DE SOFTWARE 53

ZOCO 64

PROGRAMACION DE JUEGOS (COLECCIONABLE) 31
HACER SALTAR LA BANCA
CONVIERTE TU AVENTURA EN UNA EPOPEYA

SIEMPRE HAY ALGO NUEVO

Acaba de llegar el **Amiga** a Europa y ya comienza a especularse en los EE.UU. en torno a un nuevo modelo, que podría llamarse **Amiga 2000.** Incluso se le atribuyen dos unidades de *diskette* de 3-1/2" y una de 5-1/4", así como la inclusión de la tarjeta de compatibilidad con el **PC.**

La Feria de Electrónica de Consumo (CES), que se celebra desde hace 20 años en Chicago, no ha traído en la presente edición la gran cantidad de novedades a que estábamos acostumbrados. La presencia de **Commodore** en el certamen ha sido más bien tímida, pero se aprovechó la ocasión para presentar el nuevo **Commodore 64C,** cuyo aspecto externo está en la misma línea del **C-128** y además dispone de un nuevo sistema operativo orientado a los gráficos **GEOS,** de tal manera que será capaz de seguir la tendencia de utilizar iconos, menús y ventanas. Para hacer más valioso al nuevo sistema operativo, disponible en disco también para los poseedores de los modelos **64 128,** se han lanzado paquetes de *software* tales como el **geoWrite, geoPaint** o **Desktop Utilities** (utilidades de mesa de trabajo). Parece que el **64** sigue gozando de la buena salud que muchos se niegan a reconocerle.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.º B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido ..... **Segundo título elegido** .....

Tercer título elegido ..... **Programa que te gustaría conseguir** .....

Qué ordenador tienes ..... **Nombre** .....

1.er Apellido ..... **2.º Apellido** .....

Fecha de nacimiento ..... **Teléfono** .....

Dirección ..... **Localidad** .....

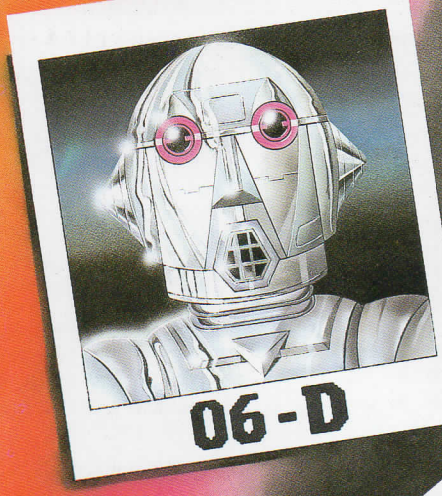
Provincia .....

OnLine®
COMPUTERS PLUS

¡¡¡entra en juego!!!

Accolade

Psi 5 Trading Co.
Una misión suicida



UNA MISION SUICIDA

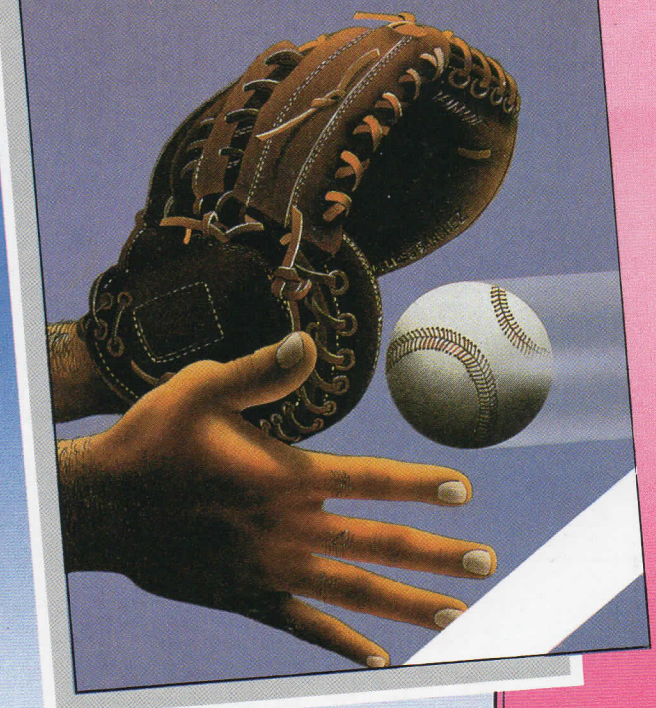
Viviendo en el filo con enemigos que parecen leer tu mente, tú eres el capitán de una nave y desde tu asiento controlas el siglo XXV. Los habitantes de PARVIN FRONTIER pagarán casi todo lo que poseen a quien pueda atravesar, con ellos, un cargamento de suministros fuera de este mundo. Esto es una misión difícil de cumplir, incluso para el experimentado capitán del PSI-5 TRADING COMPANY.

HARDBALL

Le bastarán solo cinco minutos para ver que todos los demás juegos de beisbol para ordenadores son de menor categoría en comparación con el HARDBALL.
Se puede jugar de dos maneras, una como juego de acción en el campo y otra como un juego de estrategia de entrenador, o ambas a la vez.

Accolade

HardBall



GALERIAS

Marcando estilo.

DE PELICULA

Una reciente nota de prensa confirma que Activision ha conseguido los derechos exclusivos para desarrollar programas relacionados con importantes producciones cinematográficas por venír. Entre los titulos se hallan Laberinto, coproducida por Lucas Films, Howard el Pato, Aliens y Problemones en la Pequeña China.
Por otro lado, esta firma acaba de lanzar el programa Hacker II: The Doomsday Papers, del que se dice que ofrece aún mas sorpresas que su antecesor, Hacker.

ORDENADOR FABRICADO EN ESPAÑA

No es nuestra costumbre mencionar en esta sección equipos de otras marcas. A no ser que vengan precedidos por algún mérito especial.
Este es el caso del compatible PC que fabrica Promax y comercializa Multihard. Ambas son dos firmas nacionales, que presentaron su estrategia durante el pasado Informat.
No es la primera vez que una firma española desarrolla un ordenador de esta clase. Otras iniciativas han tenido distintos grados de fortuna.
Promax es una empresa fabricante de instrumentos electrónicos de medida, que goza de excelente reputación mas allá de nuestras fronteras.
Confiamos en que este sea el principio de un fructífero camino que lleve a nuevas lineas de fabricación.

NO ERA ESE GOLF

El mes pasado nos haciamos eco del nuevo golf de Ariolasoft y dabamos como referencia la sección de Softactualidad. Pues bien, se trata de dos programas diferentes.
Golf Construction Set es un magnífico programa que permite además crear un recorrido propio de 18 agujeros, aparte de los cuatro previstos en principio. Dentro de poco lo describiremos con mayor detalle. Parece que el golf informático va a ponerse de moda.

NUEVO COMMODORE 64

A principios de junio, durante el Consumer Electronics Show, presentó Commodore su nueva versión del C-64.
Este modelo, bautizado C-64C, tiene un aspecto externo que sigue la linea del C-128 y además estrena sistema operativo.
GEOS significa sistema operativo para ambiente gráfico. No reside en el ordenador, sino que se le vuelca desde disco y dota al ordenador de iconos y menús de elección, en la misma linea de trabajo que utiliza el Workbench del Amiga. El control puede realizarse con *joystick* o ratón
La compatibilidad con el C-64 tradicional es completa. La resolución gráfica es la misma y emplea el BASIC en versión 2.0.
El GEOS dispone de programas de aplicación, tales como el procesador de textos GeoWriter, el paquete gráfico GeoPaint y el "todo uso" Desktop Utilities, que incluye calculadora,calendario, alarma y block de notas.

LOS DIEZ GANADORES DEL CONCURSO INPUT/ANAYA MULTIMEDIA

A continuación mostramos la lista de los diez elegidos entre quienes habeis respodido a la primera convocatoria del concurso.

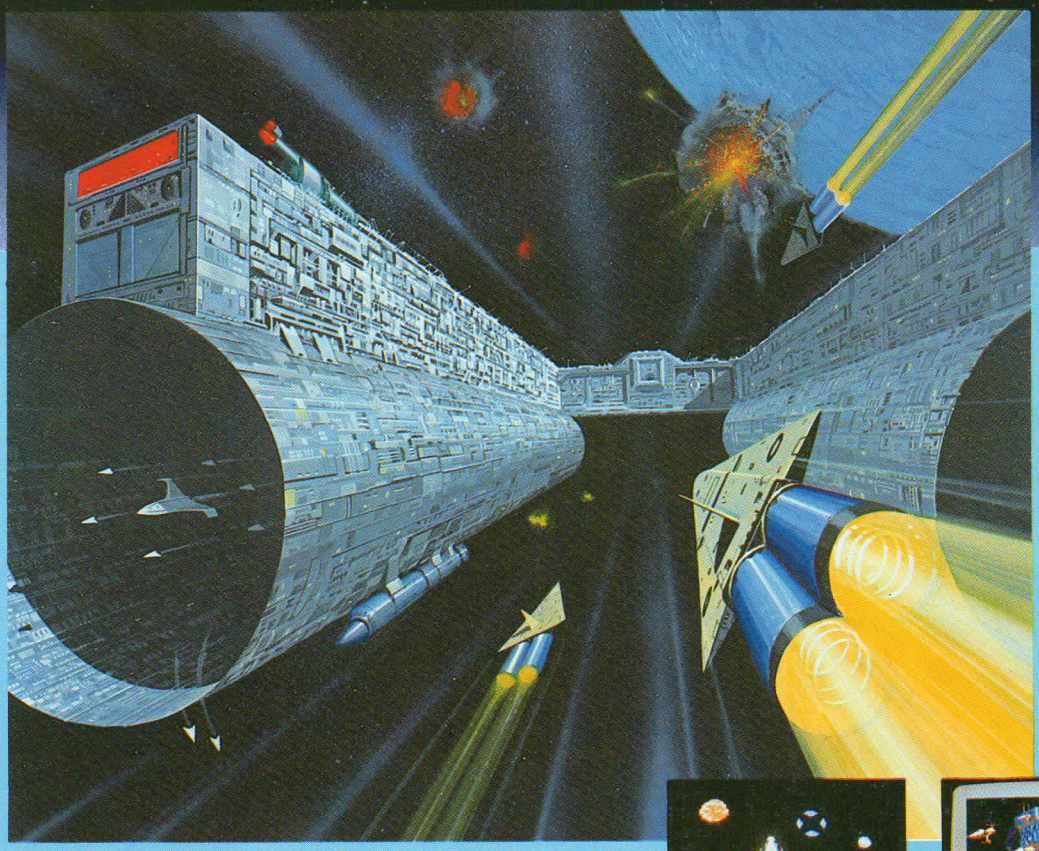
* Jose Luis Alcaraz Betoret. Sabadell. Barcelona.
* Jorge Borrego Sevilla. Madrid.
* Rafael Ayllon Diaz. Badalona. Barcelona.
* Delia Reina. Getxo. Vizcaya.
* Gustavo Adolfo Perez Agudo. Malaga.
* Encarna Ferrer Perez. Alaquas. Valencia.
* Ricardo Ferran Pedreira. Barcelona.
* Miguel Angel Herrero Matesanz. Madrid.
* J. Antonio Lopez Molina. Ugijar. Granada.
* Raul Diez Nunez. Reinosa. Cantabria.

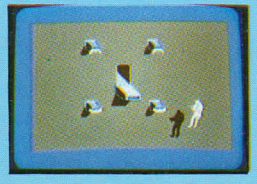
En breve plazo recibireis una carta notificandoos el premio, junto con el catálogo de titulos de la editorial para que elijais los deseados. Gracias por vuestra participación.

DESAFIANDO AL IMPERIO

Pantallas e Instrucciones en CASTELLANO

STARSHIP ANDROMEDA





ALANA, la malvada Tirana, gobierna la Galaxia despóticamente con su Super Computador Mental (S.C.M.). La única forma de acceder al S.C.M. es utilizar el Cristal-Llave que cuelga del cuello de ALANA con una cadena de durísima pentita. ¡Sólo la Lanza de protones puede cortar esa cadena!. ¡Busca la Lanza de protones por toda la Galaxia a bordo de la ANDROMEDA, sin olvidar los dos elementos rarísimos que le proporcionan energía: Trysst y Zyron!.
¡Arrebátale el Cristal-Llave y reprograma el S.C.M.!
El destino de la Galaxia está en tus manos.

Soberbio sonido y gráficos excepcionales.
12 juegos de acción y estrategia en 1.
10.000 localizaciones distintas en el Metaespacio.
Sistema de almacenamiento por palabras clave.
Un autentico desafío para la habilidad y el ingenio.
¡Un sueño más allá del Infinito!.

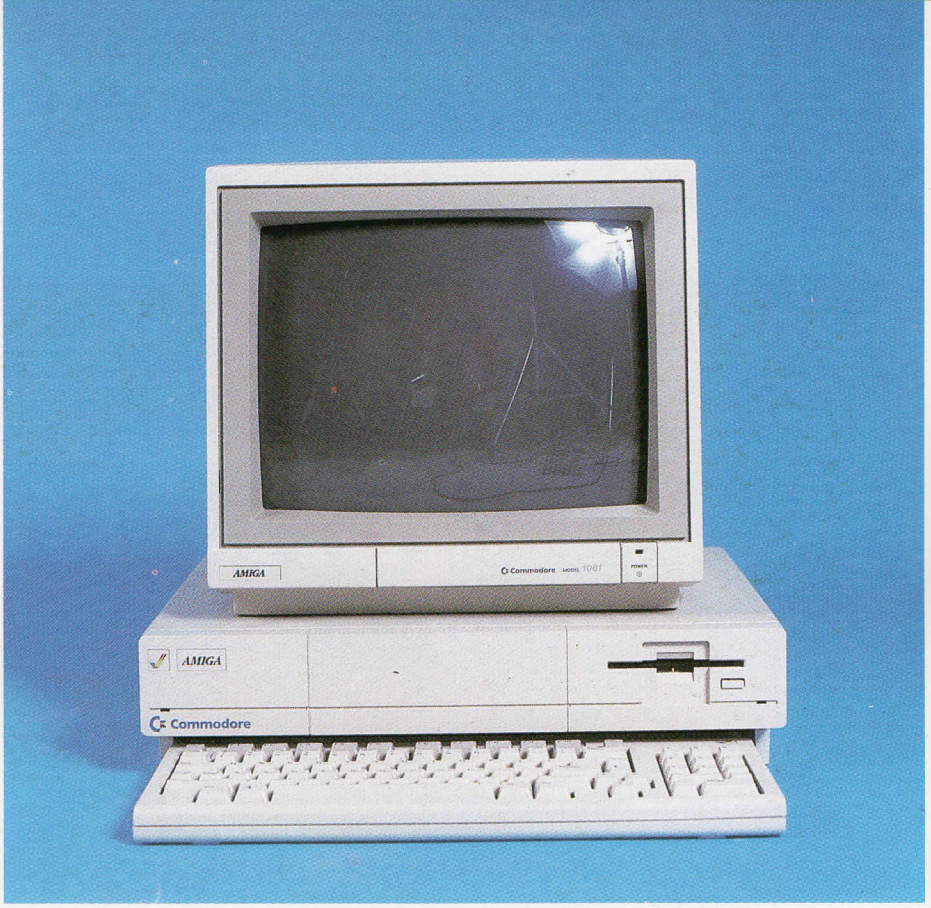
DRO SOFT

Editado por DRO SOFT. Fundadores, 3 - 28028 Madrid
Tlfs.: 255 45 00 / 09

ariolasoft

FABULOSO AMIGA

Puede que la potencia para el proceso de datos fuese el factor clave para definir la bondad de un ordenador. A partir de ahora, las capacidades para gestionar gráficos empiezan a ser un argumento definitivo en la nueva informática.



Son muchos los expertos que opinan que la industria microinformática ha errado sistemáticamente en una tarea: Convencer al ciudadano medio de lo fácil que puede ser el manejo de un ordenador. Pero tal vez ello se debiera a la carencia de un *interface* máquina-hombre humanizado.

Los menús de elecciones están bien, pero no demuestran ser suficiente para romper el hielo. Bien es cierto que la aplicación de la iconografía —imágenes aclaratorias que representan una opción— no es nada nuevo. El sistema **Star**, de **Xeros**, u ordenadores como los **Lisa** y **Mackintosh**, de **Apple**, y los modelos **ST**, de **Atari**, ya los utilizan habitualmente.

El **Amiga** acaba de desembarcar creando la misma inusitada expectación que provocara años atrás el modelo **64**. Si bien ambos ordenadores se situan en gamas diferentes.

Amiga es el nombre de la compañía que lo diseñó originalmente, siendo absorbida hace tiempo por **Commodore**. Es un ordenador que representa un gran paso adelante en lo que debe ser un ordenador personal todoterreno.

DESCRIPCION DEL EQUIPO

El ordenador **Amiga** responde a la clásica configuración modular de tres elementos fundamentales: Unidad central, teclado y monitor separados físicamente. Comparten el mismo agradable color tostado en el plástico de sus carcasas.

La unidad central tiene forma paralelepipédica y menor tamaño que la de un **PC**. Su base queda separada de la mesa por unas patas alargadas, que dan cobijo al teclado cuando no se utiliza.

La cara frontal presenta en el lado izquierdo el logotipo y la leyenda del **Amiga**. Debajo asoma por una hendidura horizontal acanalada el tímido LED verde, que indica la presencia de alimentación eléctrica en el ordenador.

En la zona derecha muestra su boca la unidad de *diskettes* de 3-1/2 pulgadas, capaz de almacenar hasta 880 Kbytes por disco, y nuevamente un LED rojo que denuncia la entrada en acción de la unidad. A su lado, en la zona central, está disponible una tapa que alojará una segunda unidad de disco. Pero de momento sólo es otra pieza de plástico que guarda armonía con el resto de la fachada.

En la parte posterior se acumulan los conectores necesarios para que el ordenador alcance el mundo exterior.

De izquierda a derecha, pasamos rápida revista. Lo primero que encontramos es el conector para el teclado, que se une al mismo a través de un cable helicoidal, similar al empleado en los teléfonos. Seguidamente tres conectores de aspecto similar y distinta aplicación: Comunicaciones en modo paralelo, acceso a unidades de *diskette* externas y comunicaciones en modo serie. A su lado aparecen los dos conectores para la salida de audio con efecto estéreo. Un cuarto conector del tipo D proporciona la salida de la señal de vídeo en distintas alternativas RGB (señales rojo, azul, verde y sincronismo por separado). Dos nuevos conectores facilitan la señal de vídeo

a un modulador de TV o, en forma de vídeo compuesto, para un monitor adecuado. Debajo, escondido en un recodo, aparece el conector para el cable de alimentación eléctrica. Finalmente, esta cara se transforma en rejilla, por la que se adivina el ventilador de refrigeración, que disipa el calor producido en el interior de la unidad central.

En el lateral derecho hay dos conectores tipo D, iguales que los empleados por el **64** para los *joysticks*. Su cometido es similar, aunque en este caso el ratón está casi permanentemente conectado en el *port* 1. Un poco más allá, una tapa alargada oculta el acceso a los *buses* del sistema y puede ser extraída para conectar futuros accesorios y periféricos. Cabe decir que la memoria RAM puede ampliarse por aquí hasta 8,5 Mbytes, aunque sospechamos que más en el futuro.

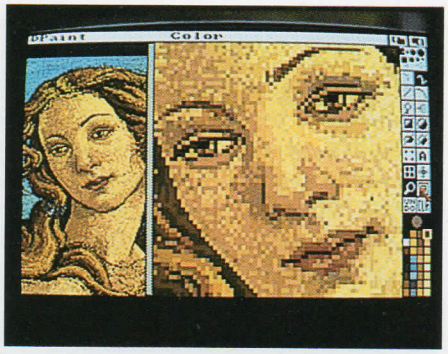
En la cara izquierda está únicamente el conmutador de encendido. Este ordenador dispone de 512 Kbytes de memoria RAM interna, aunque los primeros modelos venían con 256 K. Por otro lado existe una zona de 256 Kbytes protegida contra la escritura, donde se almacena información delicada, tal como puede ser el sistema operativo volcado desde el disco.

El teclado está compuesto por un total de 89 teclas. De ellas, 13 son un teclado separado tipo calculadora, (aunque no lleva incorporadas las operaciones matemáticas básicas) muy útil para la introducción de información numérica.

El cuerpo principal es un teclado en configuración clásica QWERTY, en el que están presentes teclas habituales como CTRL (control), CAPS LOCK (sólo mayúsculas), TAB (tabulación) o BACK SPACE (un espacio hacia atrás). Han sido incluidas otras no tan habituales, como HELP (petición de ayuda en determinados programas), ALT (juego de caracteres alternativo, utilizada en el **PC**) y A (de **Amiga**). Estas dos últimas aparecen por duplicado en ambos extremos de la barra de espacios.

En la esquina inferior derecha se han situado las teclas de control de movimiento del cursor que, aún estando agrupadas en cruz para mayor comodidad de uso, no están separadas del teclado principal.

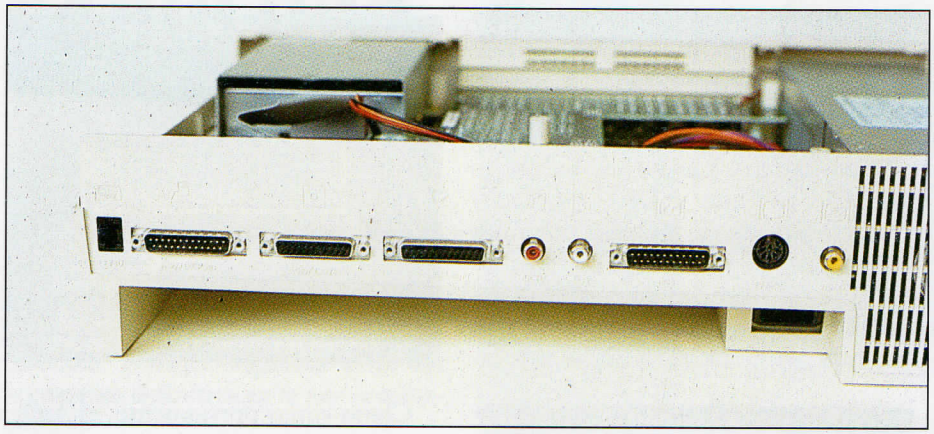
La línea superior está formada por 12 teclas que contrastan del resto por su color oscuro. Las dos de los extremos son ESCape y DEL (borrado) respectivamente y las diez centrales son dos grupos de 5 teclas de función. Por tratarse de teclas de cometido definible por *software*, en la franja que hay más arriba de ellas, se ha rebajado el plástico para alojar fácilmente etiquetas aclaratorias de la función asignada a cada una de ellas en un programa.



La carcasa del teclado incorpora un microprocesador propio, el **6500/1**, con 2 Kbytes de ROM y 64 bytes de RAM. La primera función que realiza es la de autocomprobación, denunciando la presencia de averías por el parpadeo de los LEDs de algunas teclas.

Como vimos previamente, el **Amiga** produce tres tipos de señal de vídeo. La primera es del tipo RGB, donde las señales fundamentales que componen son obtenidas por separado, proporcionando la imagen de mayor definición. La segunda responde al estándar NTSC y está compuesta por una combinación de las señales anteriores. NTSC es el estándar de televisión utilizado en los EE.UU. Sin embargo parece que de momento no hay disponibles modelos adaptados al estándar del sistema PAL, muy difundido en Europa. Una de las justificaciones que se barajan argumenta que la comercialización del **Amiga** junto con su monitor no hace necesaria la transformación.

La tercera señal es la aplicable a un modulador de UHF para utilizar un televisor doméstico, pero la diferencia en la calidad de la imagen es tan grande que no se recomienda esta solución.



GRAFICOS

En el primer momento, lo que más llama la atención en el **Amiga** es la enorme calidad de los gráficos que produce. Inmediatamente después es la increible velocidad con que manipula los objetos visualizados. El secreto reside en una habilidad del *chip* **Agnus**, llamada **Blitter** y que no es otra cosa que una abreviatura de *Block Image Transferrer* (transferidor de imágenes en bloque), trasladando el contenido de la memoria que almacena la imagen de un lugar a otro, como si se tratara de un bloque único. De esta forma libera al superveloz **68000** del manejo de las imágenes.

En modo gráfico el ordenador tiene 4 resoluciones de trabajo: 200 por 320, 400 por 320, 200 por 640 y 400 por 640 *pixels* o puntos elementales. Cuando visualiza 400 *pixels* en modo vertical está aprovechando ambos semicampos del barrido de una imagen completa de la pantalla. En efecto, el haz de electrones que forma la imagen en la pantalla de televisores y monitores, barre esta dos veces para formar una imagen. La primera vez lo hace formando líneas paralelas. En la segunda hace lo mismo, pero cada nueva línea se sitúa entre dos de las anteriores entrelazándolas, creando así una sensación de continuidad en la superficie de la imagen. Aprovechando esta característica el **Amiga** es capaz de duplicar la resolución vertical. La paleta de colores posibles se traduce en una gama de 4.096 posibles. De ellos se pueden elegir 32 simultáneamente en los modos de baja resolución (200 líneas).

Es posible utilizar un recurso del sistema, llamado **HAM** (Mantenimiento y modificación) que permite en determinadas situaciones disponer

SEINFO, S.L.
SERVICIOS DE INFORMATICA

PROGRAMAS PROFESIONALES
Commodore 64 - 128

CONTABILIDAD-128
Basada en el Plan General de Contabilidad. 1.500 cuentas contables. Fichero de apuntes ilimitado. Contrapartida automática. Recomposición de apuntes. Control de cuadre de diario. Extractos por pantalla o impresora. Balance de sumas y saldos. Balance de Situación. Cuenta de explotación. Listado de IVA soportado y repercutido. Gestiona varias contabilidades. 80 columnas con cualquier monitor.

CONTABILIDAD-64
Basada en el Plan General de Contabilidad. 300 a 1.000 cuentas. Control de cuadre de apuntes. Extractos por pantalla o impresora. Balances programables.

GESTION COMERCIAL
PAQUETE INTEGRADO DE FACTURACION Y CONTROL DE STOCKS. Ficheros programables por el usuario. Entradas/salidas de almacén. Inventario permanente y bajo mínimos. Reserva de pedidos. Facturación y emisión de recibos. Control de IVA. Listados selectivos por zonas, ventas, compras, tarifas, etiquetas, etc. 2.000 artículos, 1.000 clientes y 340 proveedores.

Estructuras ..... 25.000 Ptas.
Calcula pórticos planos de hormigón armado. Calcula los esfuerzos para las tres hipótesis verticales, viento y sismo. Armado total de vigas y pilares. Cuadro de pesos de hierro. Cuadro cúbico de hormigón. Listado de todos los esfuerzos en el armado.

Mediciones y presupuestos 25.000 Ptas.
Programa de mediciones y presupuestos de obra totalmente programable por el usuario. Listado de mediciones y presupuesto por partidas. Posibilidad de ajuste automático de presupuesto.

Facturación ..... 15.000 Ptas.
Programa de facturación directa. Fichero de artículos y clientes. Diarios de ventas. Desglose de impuestos. Emisión de recibos. Varias versiones.

Control de Stocks ..... 15.000 Ptas.
Ficheros de artículos y proveedores. Control de entrada/salida de almacén. Actualización automática. Inventario permanente, Inventario bajo mínimos. Listados varios.

Tratamiento de textos ..... 5.850 Ptas.
Teclado castellano. Fácil manejo.

(Pida información sobre otros programas profesionales y técnicos)

Pida información **SEINFO, S. L.** Avda. Goya, 8 - 50006 ZARAGOZA - Teléfs. (976) 22 69 74 - 23 29 61

de los 4.096 colores simultáneamente en la misma pantalla. El **HAM** puede explicarse como el troceo de las líneas barrido por secciones, cambiando rápidamente el valor del color del *pixel* afectado en cada instante.

El **Amiga** puede manejar hasta ocho *sprites* de cualquier altura pero la anchura se ve reducida a tan sólo 16 *pixels*.

SONIDO

El *chip* bautizado como **Paula** se ocupa también de dotar al ordenador de unas capacidades sonoras difícilmente presentes en un microordenador. Existen cuatro voces de sonido en modo estéreo, pudiendo igualmente reproducir sonidos previamente digitalizados por el ordenador.

Las cuatro voces pueden llegar a ser hasta 16 por cada uno de los dos canales cuando se trabaja en multitarea.

La forma de la envolvente no está predeterminada por una reducida gama de opciones, como en el **64**, sino que es factible definirlas a voluntad.

La frecuencia de trabajo de cada voz se extiende a lo largo de 9 octavas.

PUESTA EN MARCHA

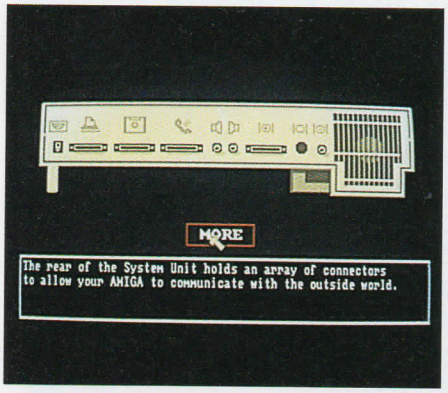
Cuando activamos el conmutador de alimentación, tras unos breves instantes en los que el sistema se autocomprueba, aparece en la patnalla el dibujo de un *diskette* con la leyenda **Kickstar**. Insertando ese disco, comienza el volcado del sitaema operativo en una zona protegida de la RAM.

La ventaja de que el sistema operativo no resida en ROM es que si en el futuro aparecen sistemas operativos diferentes, solamente será necesario disponer del nuevo *diskette* para que el ordenador lo adopte inmediatamente. Después pide que insertemos el disco **Workbench**.

Lo primero que visualiza el **Workbench** es una ventana con una banda blanca horizontal en la zona superior. En ella aparece el título del *diskette*, su versión, la cantidad de memoria disponible y dos pequeños dibujos, constantes en todas las ventanas que aparecerán. Sirven para decidir en qué momento aparece una ventana encima (primer plano) o tras otra (fondo).

Sólamente con llevar el cursor a una de ellas y presionar la tecla del ratón toma protagonismo la ventana elegida. También aparece dibujado un icono representativo del *diskette*, con la leyenda **Workbench** bajo él.

En la versión actual del **Workbench**



existen 4 cajones, cuyos iconos van etiquetados como sigue: Demostraciones, Utilidades, Sistema y Vacío.

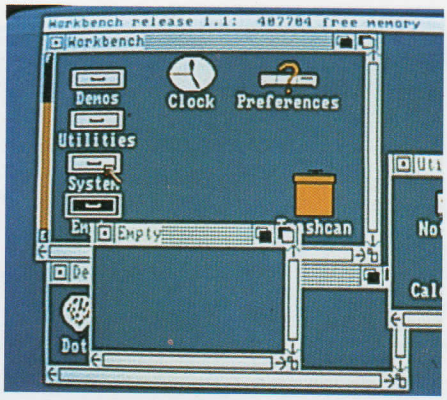
Demos son tres programas que ponen de relieve las capacidades multitarea del **Amiga**.

Utilidades visualiza una calculadora en la pantalla, con sus teclas y todo, y además aporta un cuaderno de notas.

Sistema incluye un copiador de *diskettes*.

El último cajón —**Vacío**— nos permite abrir un fichero propio que formará parte del **Workbench**.

Todo ordenador debe disponer de un sitema operativo que se encargue

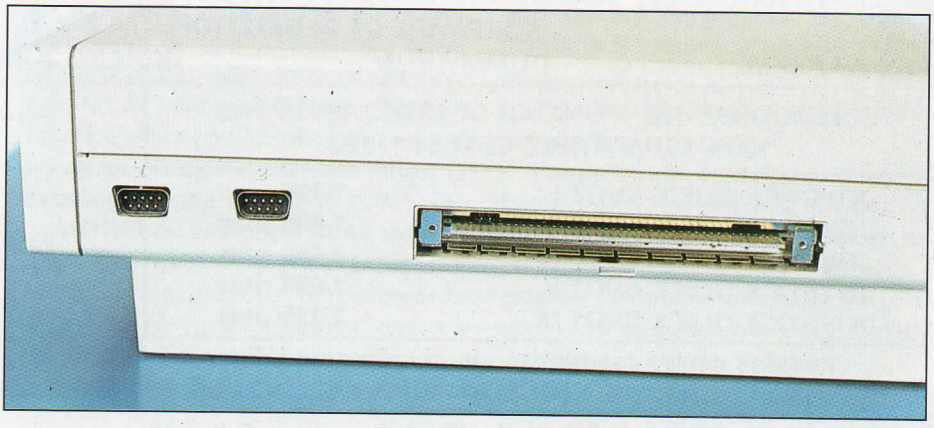


de que el ordenador trabaje por si mismo en multitud de pequeñas tareas elementales. El **Amiga** dispone de dos sistemas operativos, que permiten trabajar con los iconos y el ratón o bien con el teclado, respectivamente.

Intuition es el nombre del *software* que maneja las ventanas, creando un *interface* máquina-hombre de naturaleza iconográfica.

AmigaDOS es la segunda opción, un sistema operativo multitarea. Cuando trabajamos con el **Workbench** se puede acceder al sistema operativo **AmigaDOS** abriendo la ventana CLI, después de haber seleccionado el icono denominado **Preferencias**. CLI significa Intérprete de Comandos en Línea y permite que utilicemos el teclado en sustituición del ratón. Como veremos en el siguiente párrafo de este artículo, **Preferencias** permite que el usuario personalice su ordenador. Pero no es necesario repetir las operaciones siempre que se ponga en marcha, es posible almacenar dichas elecciones en el disco. Esto implica también al CLI.

Preferencias es una herramienta



que permite al usuario efectuar determinados cambios en el funcionamiento normal del **Amiga**. Se puede variar la velocidad de autorrepetición de las teclas, poner en hora y fecha el reloj, personalizar el manejo del ratón, elegir el tipo de impresora, etc... y hasta decidir la relación entre cada uno de los tres colores básicos que formarán las imágenes del **Workbench** en la pantalla.

Con el **Workbench** la apertura y cierre de ventanas de trabajo es tan sencilla como llevar el cursor a la banda blanca de la línea superior y pulsar las teclas del ratón.

La lectura del contenido de *disquettes* y su carga en el ordenador es tan fácil que podemos decir que es intuitiva (tal vez de ahí el nombre del *interface* por *software*). Una nota simpática, cada vez que se inserta un *disquette* en la boca de la unidad, es leído su nombre inmediatamente, apareciendo un nuevo icono en una esquina de la pantalla.

Entre el *software* que viene acompañando al equipo, existe un breve curso de aprendizaje para manejar el ordenador. Realmente parece un cursillo audiovisual, en el que la siguiente imagen aparece por fundido de la anterior. Su manejo es totalmente interactivo. Esta es una breve muestra de lo que puede ser la enseñanza asistida por ordenador en el futuro inmediato, si alguien se encarga de preparar los programas, claro está.

Los comandos del **AmigaDOS** son 45 de sintáxis sencilla y algunos recuerdan a los utilizados en el **MS DOS**. Se encuentran almacenados en el subdirectorio C del *diskette* **Workbench** y por tanto no son residentes en el ordenador, sino que deben ser cargados en la RAM protegida para utilizarlos.

El sistema de manejo de ficheros del **AmigaDOS** es sorprendente, permitiendo agrupar ficheros comunes en subdirectorios independientes y definir nuevos subdirectorios dentro de ellos. De esta manera es más fácil analizar los contenidos de un *diskette*. Es aquí donde el concepto de cajón, asociado a directorio, toma cuerpo. Se crea una especie de estructura jerárquica, que permite el agrupamiento de ficheros en distintos subdirectorios.

Una de las ventajas inmediatas consiste en la posibilidad de utilizar el mismo nombre para ficheros situados en subdirectorios distintos.

Otra de las muchas posibilidades del **AmigaDOS** consiste en la creación de discos de RAM, que son zonas de la memoria que el sistema utiliza como si se tratase de discos de acceso muy rápido.

El **AmigaDOS** recuerda al sistema operativo **Unix**, pero simplificado. No es de extrañar, puesto que existen versiones para el **68000**. Por otro lado el **AmigaDOS** es más o menos el S.O. **Tripos**, muy relacionado con el **Unix** y para un solo usuario (por contra a multiusuario).

El BASIC estándar que utiliza este ordenador está bautizado, como no,

EL IVA LO PAGA MICRO-1

MICRO-1

C/ DUQUE DE SESTO, 50
28009 MADRID
METRO O'DONNELL O GOYA
APARCAMIENTO GRATUITO EN FELIPE II

SOFTWARE: ¡¡2 PROGRAMAS POR EL PRECIO DE 1!!

CAULDRON	2.500 ptas.	SABOTEUR	2.500 ptas.
«V»	2.500 ptas.	PING PONG	2.500 ptas.
ROCK'N LUCHA	2.500 ptas.	YIE AR KUNG FU	2.300 ptas.
URIDIUM	2.500 ptas.	RAMBO	2.300 ptas.
DINAMITE DAM	2.100 ptas.	GOONIES	2.600 ptas.
ZORRO	2.100 ptas.	EXPLODING FIST	2.100 ptas.
FIGHTING WARRIOR	2.100 ptas.	BEACH HEAD II	2.600 ptas.
MILLION II (BRUCE LEE, MATCH DAY, MATCH POINT, ENTOMBED)			2.500 ptas.

SOFTWARE DE REGALO (OFERTA 2 × 1)

PSY WARRIOR — DUMMY RUN — INDIANA JONES — POLE POSITION

COMMODORE 128 + UNIDAD DE DISCO 1571 PTAS.
100% COMPATIBLE CBM 64 113900

¡¡20% DE DTO. SOBRE P.V.P. EN IMPRESORAS!!

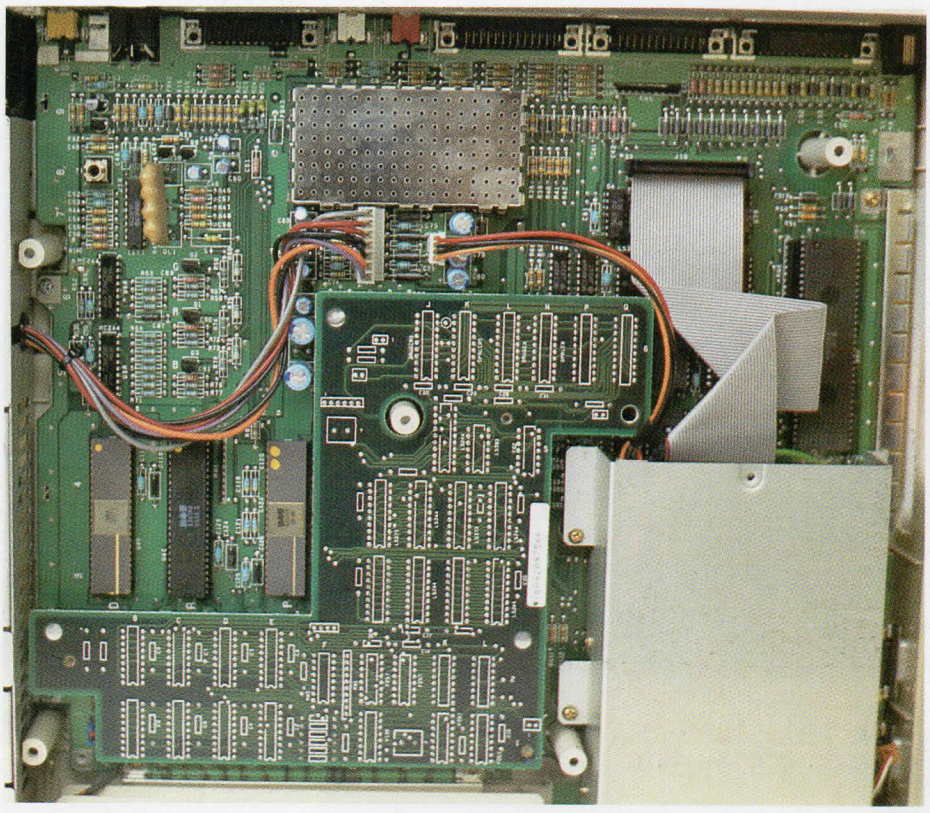
JOYSTICK QUICK SHOT I	1.395 ptas.
JOYSTICK QUICK SHOT II	1.695 ptas.
JOYSTICK QUICK SHOT IV	2.490 ptas.
JOYSTICK QUICK SHOT V	1.695 ptas.
JOYSTICK QUICK SHOT IX	2.395 ptas.

BUSCAMOS DISTRIBUIDORES
¡¡GRANDES DESCUENTOS!!

Pedidos contra reembolso sin ningún gasto de envío. Teléfs. (91) 275 96 16, o escribiendo a:
MICRO-1 - C/ Duque de Sesto, 50 - 28009 MADRID

con el nombre de **Amiga BASIC**. Ha sido desarrollado por **Microsoft**, la firma más prolífica en la puesta en marcha de versiones del BASIC (el sistema operativo **MS DOS** también es obra suya). El editor de programas es una pieza clave para la comodidad en la programción y en este caso podemos asegurar que se han roto los moldes de la versatilidad.

El **Amiga BASIC** incorpora un editor de objetos, que permite crear *sprites* y animación de manera sencilla. Una aportación realmente interesante es la programación estructurada (al BASIC se le ha venido acusando de carecer de capacidades para este tipo de programación). El editor permite indentar las líneas de programa, como por ejemplo en el **Pascal**, que ofrecen mayor facilidad para identificar las rutinas, que a su vez se pueden etiquetar. Igualmente se pueden crear subprogramas externos y que todo ello funcione como conjunto.



RESUMEN

El **Amiga** no es sólo un ordenador de gestión y tampoco una excelente maquinita de juegos. Es ambas cosas y mucho más.

Una de las definiciones que más se aproximan al resumen de lo que el **Amiga** es, viene a decir algo así como que se trata de un sistema multitarea, capaz de ejecutar diversos cometidos simultáneamente.

LOS CUATRO MAGNIFICOS

Los motores que proporcionan su formidable potencia de proceso al **Amiga** son cuatro *chips*: **MC68000**, **AGNUS**, **DENISE** y **PAULA**.

El primero es un microprocesador más o menos estándar, mientras que los tres restantes son de diseño exclusivo a medida de este ordenador.

El **68000** no es de reciente aparición y ya en el 82 **Radio Shack** lo incorporó en su adelantado ordenador **Tandy III**, que también se dejó ver en nuestro país. Lo interesante de aquel modelo era que tenía como *partenaire* al microprocesador **Z80** de 8 bits. Fue el primero en incorporar dos microprocesadores. Después el **68000** pareció caer en el olvido, siendo repopularizado por **Sinclair**, que incorporó a un miembro de esta familia en su **QL**.

Realmente el **68000** no es un único, sino una familia de microprocesadores con sus respectivos *chips* de soporte. Siempre que se habla de él no tarda en surgir la polémica. Unos dicen que su arquitectura es de 16 bits y otros que de 32. Pero parece que se llega a un acuerdo al decir que es un microprocesador de 16/32 bits, siendo capaz de direccionar algo más de 16 Megabytes de memoria externa. Internamente trabaja con 32 bits, pero los *buses* de direccionamiento que llegan al exterior del *chip* son de 16 vías.

AGNUS es una abreviatura de *Address Generator Chip* (*chip* generador del direccionamiento). Se encarga de coordinar todos los accesos directos a la memoria (no a través del microprocesador) para buscar o depositar información. Para ello se utiliza el término «canales» y existen 25 de ellos dedicados a cuestiones especializadas.

El **AGNUS** incorpora un coprocesador bautizado como **COOPER**, que se ocupa de la correcta sincronización de la señal de vídeo en la pantalla.

El **AGNUS** dispone de circuitería capaz de transferir con la rapidez del rayo bloques de información entre distintas zonas de la memoria del **Amiga**. A este dispositivo se le ha denominado **BLITTER** (*Block Image Transferrer*).

DENISE viene de *Display Encoder Chip* (codificador de lo visualizado) y se ocupa de la creación de la imagen que aparece en la pantalla, almacenando la información necesaria de un modo codificado que facilita el manejo del color. También maneja la información de los *sprites*.

Por otro lado, este *chip* es quien comunica directamente con el ratón.

El *chip* **PAULA** es el verdadero responsable de las comunicaciones con el exterior. Controla los *ports*, la generación de audio y hasta a las unidades de disco.

Estos tres circuitos integrados actúan como verdaderos coprocesado-

res, descargando a la CPU de cuestiones especializadas que le restarían rapidez.

WORKBENCH

Workbench significa banco de trabajo. Es un programa incorporado en un *diskette* y es capaz de exprimir el potencial del **Amiga** haciéndolo de fácil manejo para el usuario.

La pantalla con ventanas multiples ofrece un aspecto muy familiar con respecto al **Mackintosh** y los **ST**, aunque esta vez se ha incorporado color.

En la práctica el **Workbench** se olvida casi por completo del teclado, cediéndole el control al ratón. Así, manejando sus dos botones junto al desplazamiento por la superficie de la mesa, es posible situar el cursor/flecha en cualquier punto de la pantalla, eligiendo entre las opciones propuestas.

Tal vez lo más importante del **Workbench** es que permite al usuario personalizar su ordenador.

Cada vez que insertamos un nuevo *diskette* en la unidad es leído su título. A continuación aparece en la pantalla una pequeña representación iconográfica del *diskette*, etiquetado con su nombre. Una vez que le «abrimos» una ventana aparece en ella su contenido. Pueden tener distintas cosas: Proyectos, dibujos, herramientas, cajones, discos o baúl de desperdicios.

En la ventana principal se puede elegir uno cualquiera de estos iconos, mediante el ratón, e inmediatamente aparece una nueva ventana que muestra su contenido.

Cada ventana puede ser reubicada con el ratón, y también es posible alterar su tamaño con la misma facilidad.

El **Workbench** hace que el manejo del ordenador no presente dificultades adicionales a los usuarios no familiarizados con él.

EL BLITTER

El **Blitter** es un dispositivo encargado de transferir con gran rapidez el contenido de un bloque de memoria a otro. La velocidad de trabajo del **Blitter** es tan grande que podría trabajar sin problemas con un hipotético **68000** que trabajase a 70 MHz (en realidad trabaja a menos de 8 MHz en el **Amiga**).

El control de las ventanas que aparecen en pantalla del **Amiga** son un claro ejemplo del **Blitter**. Su tamaño y proporciones son definibles en cualquier momento, así como la ubicación o si quedan por debajo o encima de otras ventanas presentes. Teniendo en cuenta la cantidad de memoria que necesita cada ventana, es fácil darnos cuenta de la cantidad de información que ha de procesarse para manipular una ventana. Simplemente haciendo unas sencillas cuentas no acabamos de salir del asombro, al comprobar que no existen molestas demoras de tiempo en la actualización de la pantalla.

El **Blitter** continúa siendo una caja de sorpresas. Veamos un ejemplo, cuando una imagen es el resultado de otras situadas en distintos lugares, el **Blitter** es capaz de recoger información procedente de tres fuentes distintas y combinarlas para que salgan por un cuarto canal hacia el lugar de la memoria de destino. Además, esos bytes pueden ser manipulados entre sí mediante operaciones lógicas: Or, And, Or Exclusiva e Inversión, estableciendo prioridades.

Una de las posibilidades inmediatas del **Blitter** es la creación de *sprites*, que se desplazan por la pantalla sin interferir el contenido del fondo.

El **Blitter** también es utilizado para el relleno de áreas delimitadas en la pantalla, sea con patrones definidos previamente o simplemente con color. Lo mismo hace con el trazado de líneas.

Esta transferencia de bloques de información por toda la memoria no es aplicable únicamente a la zona destinada a la pantalla.

ALGO DE SOFTWARE

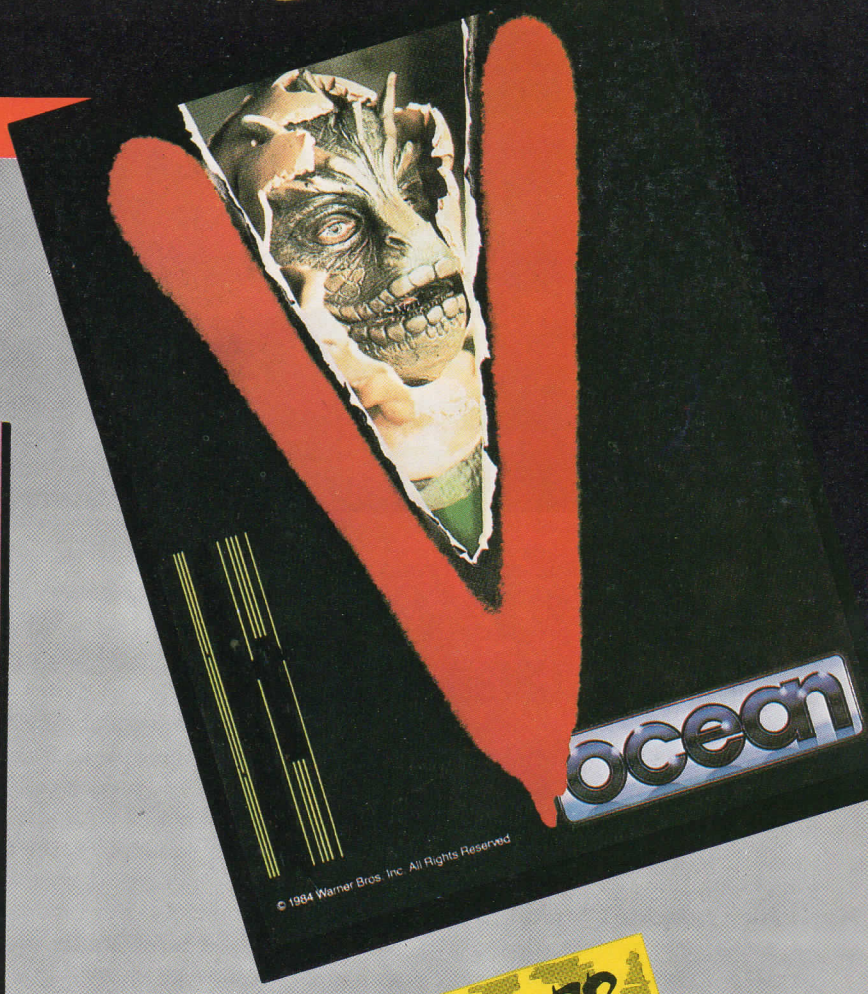
De antiguo saben los fabricantes de *hardware* que no importa lo buenos que puedan ser sus ordenadores si no existe un buen soporte para ellos. Sin una nutrida biblioteca de programas, no tienen mucho que hacer.

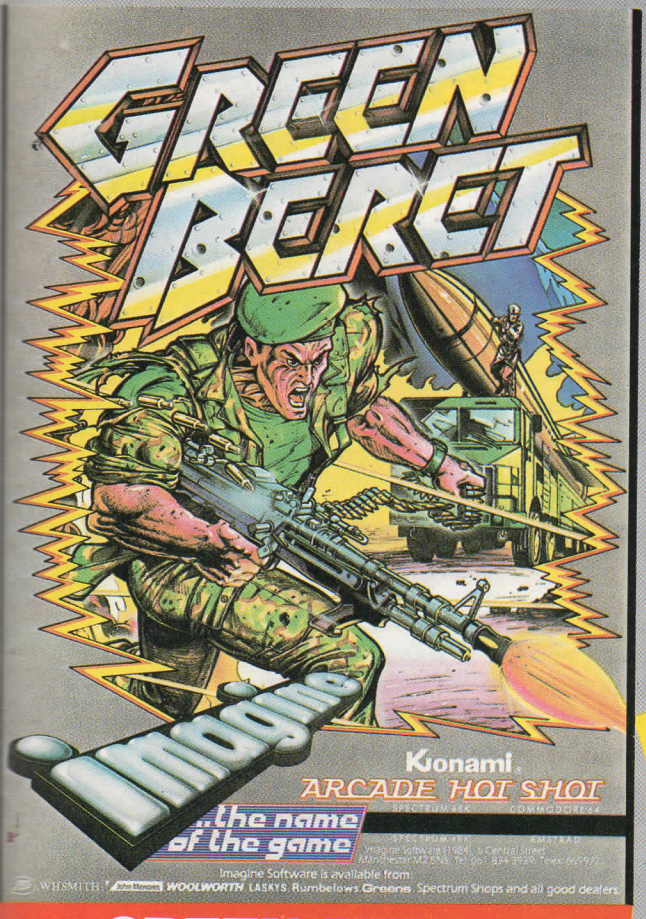
Son varias las casas de *software* que dispusieron de prototipos tempranos del **Amiga** y han trabajado duro de cara al lanzamiento.

Erbe dades

"V"

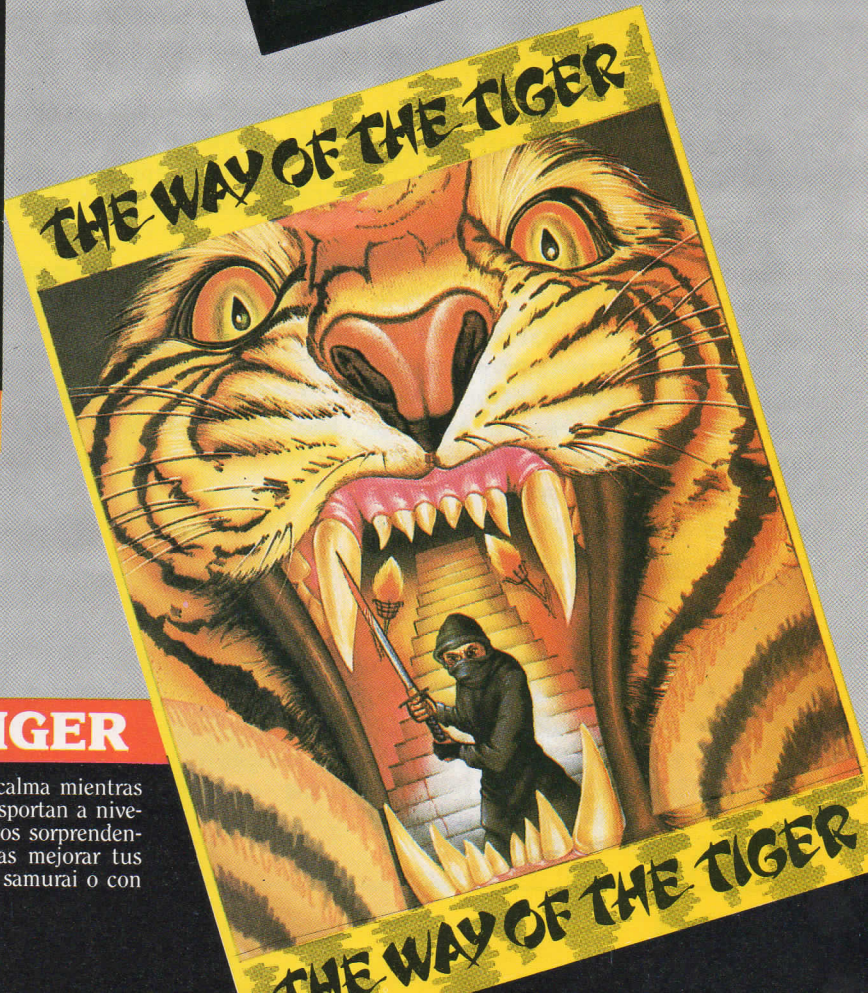
La serie de ciencia-ficción televisiva invade tu ordenador. La Tierra es visitada por alienígenas de aspecto humano. Si embargo, estos reptiles pronto revelarán su objetivo final... conquistar la Tierra y esclavizar a la raza humana.





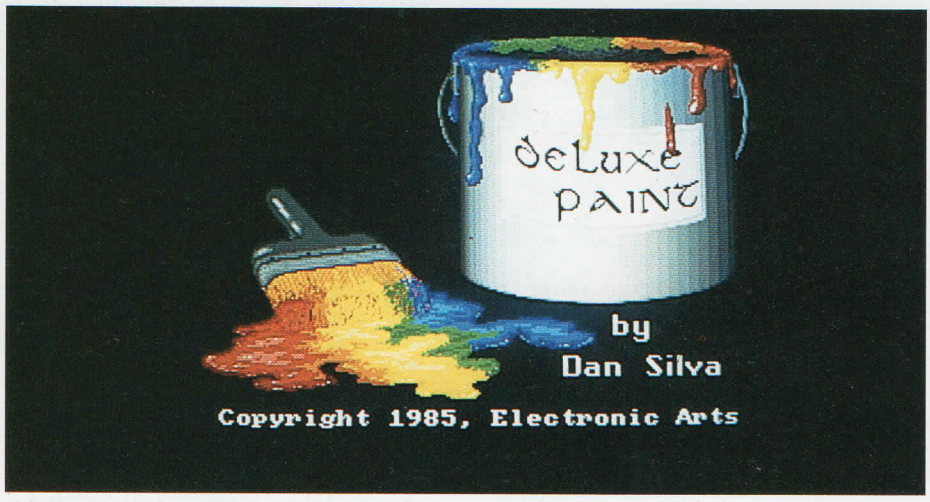
GREEN BERET

Green Beret (boina verde), es el programa de mayor acción que hayas visto jamás en un ordenador.
Decir Green Beret es decir un hombre perfectamente entrenado para la lucha. Prepárate...



THE WAY OF THE TIGER

Entra en el mudo de los samurais. Mantén la calma mientras el movimiento y las rutinas de combate te transportan a niveles que nunca pensaste posibles. Experimenta los sorprendentes efectos del "Triple Scroll" mientras intentas mejorar tus técnicas de lucha cuerpo a cuerpo, con espada samurai o con mil posibilidades más.



Electronic Arts es una de las firmas que se pusieron pronto manos a la obra para desarrollar programas de altura. El **Deluxe Paint** es un paquete sofisticado de diseño gráfico llamado a convertirse en estándar obligado. Es el primer programa lanzado capaz de demostrar el potencial del **Amiga** en el área de los gráficos.

Otros paquetes de **E.A.** son el **Deluxe Print**, que convierte al ordenador en una verdadera imprenta, y el **Deluxe Video**, que permite desarrollar viñetas e historietas gráficas por uno mismo.

En otro área que suscita gran interés en los EE.UU. ha desarrollado el **Financial Cookbook**, libro de cocina financiero, que ayuda a llevar las finanzas personalmente.

Algunos de los excitantes juegos de **Electronic Arts** han visto igualmente la luz, tales como el **Skyfox** (caza de combate), **Archon** (complejo ajedrez entre la luz y la sombra), **One-on-One** (baloncesto), **Regreso a Atlantis** (aventura submarina tridimensional), **Las siete ciudades de oro** (tras los pasos de los descubridores) o **Artic Fox** (increíble simulador de vuelo). Los potenciales sonoros del **Amiga** son descubiertos por **Instant Music**.

Por su parte, **Activision** ha actualizado algunos de sus éxitos del **Commodore 64**, mejorándolos en la medida de las lujosas aportaciones técnicas del **Amiga**. Así, juegos como el **Hacker**, **Mind Shadow** y **Alguien vive dentro de mi ordenador** (**Little Computer People**) asombrarán a quienes los admiraron en el **64**. Por otro lado, el paquete **The Music Studio** promete ser algo asombroso cuando esté a la venta.

Existe, además, un emulador del sistema operativo **MS DOS**, que permite al **Amiga** ejecutar programas desarrollados para el **PC** de **IBM**.

FICHA TECNICA

* Microprocesador **68000** desarrollado por **Motorola**. Arquitectura mixta 16/32 bits. Registros internos de 32 bits y *bus* de datos de 16 bits. Trabaja a 7,16 MHz.

* RAM central de 512 K, extensible hasta 8,5 Mbytes.

* 3 coprocesadores para tareas especializadas.

* Memoria ROM de 256 Kbytes. En realidad es RAM protegida contra escritura.

* Teclado separable, con 89 teclas divididas en las siguientes áreas: QWERTY, numérica, función y cursor. Incorpora microprocesador propio.

* Formatos de pantalla:

40, 60 u 80 caracteres por línea en modo texto y 25 líneas.

* Resolución gráfica:

Modo baja resolución: 200 por 320 y 400 por 320 puntos.

Modo alta resolución: 200 por 640 y 400 por 640 puntos.

Paleta de 4.096 colores. 32 colores en modo baja resolución y 16 en alta. Posibilidad de utilizar los 4.096 con la técnica HAM.

* Salida de vídeo con variantes: RGB (analógica o digital) y vídeo compuesto. Conector para modulador de televisión. Utiliza estandar NTSC de momento.

* Salida de audio por dos canales estéreos. Total de 4 voces en 9 octavas. Envolvente programable. Síntesis de voz.

* Ratón de control.

* Unidad de *diskettes* de 3-1/2 pulgadas con capacidad para almacenar hasta 880 Kbytes por disco.

* *Port* serie programable.

* *Port* paralelo programable.

* Dos *ports* para *joystick* o controladores (ratón, p. ej.).

* *Port* de expansión, capaz de admitir hasta tres unidades de *diskette* externas de 3-1/2 ó 5-1/4 pulgadas.

* Conector para cartuchos, periféricos, RAM o disco duro.

TU PUEDES SER EL AFORTUNADO

Si quieres participar en el concurso promovido por Erbe Software, recorta y pega esta solapa para ir completando la página que te permite acceder al sorteo.

Julio

SOUND SAMPLER, MUSICA AL BYTE CUADRADO

Tu voz puede sonar grave, como la de un enorme gigante devorador de criaturas, de la misma forma que se transforma en lo más parecido al inconformista tono agudo del Pato Donald**. Sin mayor complicación puedes demostrar que en el interior de tu ordenador se esconde una caja de ritmos y una guitarra eléctrica.**

La manipulación electrónica del sonido era, hasta hace poco, un campo casi exclusivamente relacionado con los complejos estudios de grabación. Sin embargo, en los últimos tiempos han aparecido algunos dispositivos que acercan estas posiblidades a la mano del usuario del microordenador doméstico, antes inalcanzables a un precio asequible. De ello es buena muestra el **Sound Sampler**, que vamos a pasar a descubrir.

Este dispositivo nos asombró el pasado año en el **Commodore Show**. Un equipo de músicos profesionales había preparado una cinta magnetofónica con una agradable musiquilla, partiendo del **Sound Sampler**, un programa de síntesis musical, un magnetofón de varias pistas y las voces de los cantantes. El resultado era francamente sorprendente, al menos para quienes no disfrutamos de oido de melómano.

Por tratarse de un producto **Commodore**, sospechamos que se lanzaría «a bombo y platillo». Sin embargo no ha sido así, aunque es posible encontrarlo en algún establecimiento especialista en instrumentos musicales.

¿QUE ES UN DIGITALIZADOR?

La Naturaleza maneja habitualmente magnitudes continuas; es decir, entre dos valores puede haber una infinidad de ellos. Por ejemplo, entre uno y cero existen tantos valores como puedan representarse por un cero seguido de cuantos decimales queramos. Por el contrario, las señales que maneja un ordenador son uno o cero, combinándose en un determinado orden para representar cualquier número deseado. Es aquí donde surge el problema: Las señales naturales son analógicas y el ordenador maneja cantidades discretas. Por tanto es preciso recurrir a un método que permita introducir las señales del exterior al ordenador.

La digitalización no es otra cosa que la toma de muestras (muestreo) partiendo de una señal, transformándolas en series de unos y ceros.

Existen diversos procedimientos de digitalización. Pero el más sencillo, y que por ahora más nos interesa, consiste en muestrear de una señal analógica eléctrica a intervalos regulares, controlados por un reloj extremadamente preciso. Después se procesa la muestra, convirtiéndola en un número binario igual a la amplitud instantánea de la misma.

Para la digitalización de sonido se utiliza como primer eslabón un sencillo micrófono, que traduce las variaciones de presión que llegan a su membrana en una señal eléctrica de caracter analógico.

El muestreo continuado produce una ristra de cifras que pueden ser almacenadas en la memoria del ordenador para su posterior tratamiento.

ASPECTO FISICO

El **Sound Sampler** comparte características de cartucho y accesorio. Sus circuitos están alojados en el interior de una carcasa de plástico de agradable aspecto y geometría irregular.

De la parte posterior surge una especie de caja, semejante al tramo final de un cartucho, por donde asoma el habitual peine de pistas de cobre en ambas caras de la tarjeta del circuito impreso. Por aquí se efectua la conexión al *port* de expansión de **C-64**. Esta es la única unión que necesita el **Sampler** para funcionar. El afianza-

COMMODORE

Nombre	P.VPC
(CART) FAST LOAD	4500
(CART) GAME KILLER	2960
(D) TIME TUNNEL	2598
(D) BOULDERDASH	2877
(D) COMMANDO	2500
(D) EIDOLON	2995
(D) ELITE	2829
(D) EXPLODING FIST	2600
(D) GOONIES	2700
(D) KUNG FU MASTER	2877
(D) LITTLE COMPUTER PEOPLE	2877
(D) RAMBO	2250
(D) REVS	3372
(D) SKYFOX	3210
(D) SUMMER GAMES II	2600
(D) SUPER ZAXXON	2764
(D) THEY SOLD A MILLION	2795
(D) URIDIUM	2250
(D) WINTER GAMES	3450
(D) YIE AR KUNG FU	2400
ARCADE HALL OF FAME	2128
CAULDRON II	1799
COMMANDO	1700
CRITICAL MAXX	1700
DESERT FOX	1900
EIDOLON	2056
ELITE	2571
EXPLODING FIST	1900
GOONIES	1990
HARD BALL	1900
HOCUS FOCUS	1799
KUNG FU MASTER	1900
LAW OF THE WEST	2162
LITTLE COMPUTER PEOPLE	2300
NEXUS	1799
PITSTOP II	2437
PSY 5	2245
RAMBO	1700
REGRESO AL FUTURO	1990
REVS	2895
SABOTEUR	1685
SCALEXTRIC	2135
SKYFOX	2400
SOUTHERN BELLE	1569
STARQUAKE II	1799
SUMMER GAMES	2410
SUMMER GAMES II	1990
THEY SOLD A MILLION II	1990
THEY SOLD A MILLION	2135
THIRTY GAMES	1930
TIME TUNNEL	1900
URIDIUM	1780
V	1750
VIERNES 13	1750
WARPLAY	1673
WINTER GAMES	1990
YIE AR KUNG FU	1990
Z	2115
ZORRO	2100

ESTOS PRECIOS NO INCLUYEN EL IVA

NUESTRAS OFERTAS SON NUESTROS PRECIOS

ENVIANOS A
SOFT INVADERS ORENSE 6 28020 MADRID
DESEO RECIBIR CONTRA-REEMBOLSO LOS SIGUIENTES JUEGOS:

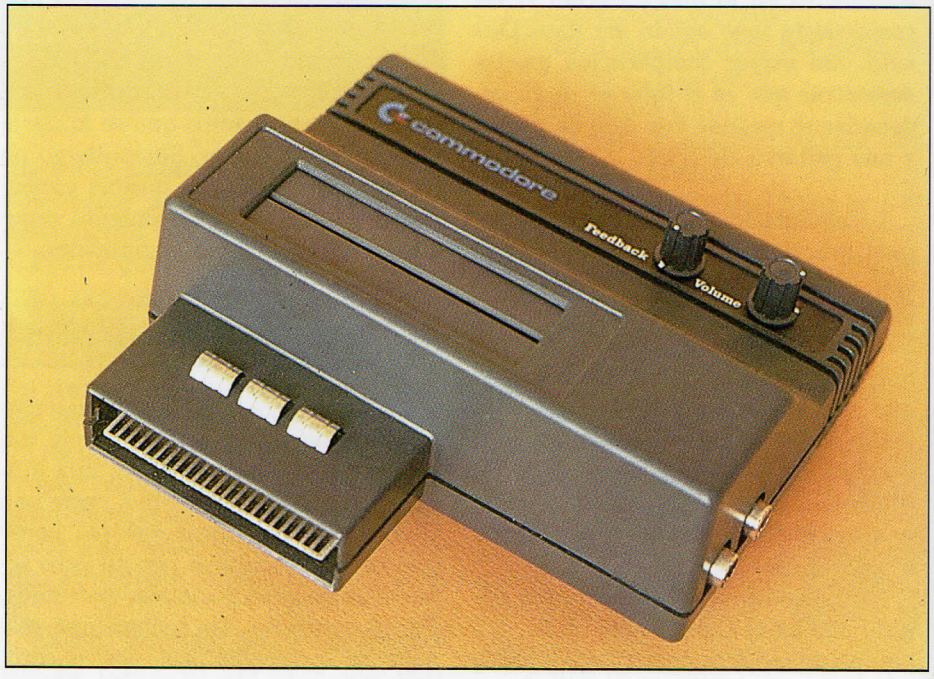
NOMBRE	CANT.	PRECIO

TOTAL + 175 (gastos envío)

ME LOS ENVIAN A: _____
NOMBRE: _____
DIRECCION: _____
POBLACION: _____ COD. POSTAL: _____
TELEFONO: _____
FIRMA

miento mecánico resulta robustecido por tres piezas metálicas que ejercen presión contra el alojamiento destinado a los cartuchos. La parte frontal presenta dos botones de control que, al girar, permiten ajustar el volumen de la señal de partida y la cantidad de realimentación aplicada en el amplificador de entrada.

A la izquierda aparecen dos conectores hembra, que tiene por misión recibir y dar salida a la señal de audio. En el primer caso se utiliza el micrófono de bajo costo que viene con el equipo, aunque puede emplearse cualquier otra fuente (por ej. un *cassette*). La señal de salida puede inyectarse en la patilla correspondiente del conector de Audio/Vídeo que tiene el **C-64**, empleando el conector que también acompaña al equipo. Por tanto se puede elegir entre escuchar el sonido a través del televisor o aplicarlo directamente al amplificador de la cadena musical. Como curiosidad, ambos conectores son del tipo coaxial empleado para las conexiones de vídeo, en lugar de los clásicos *jacks* utilizados en audio, pero esto no presenta ninguna complicación adicional. En la parte superior ha sido dispuesta una ranura con tapa escamoteable de doble hoja, que permite la introducción de otro cartucho cuando no se utiliza el **Sound Sampler**, evitando desconectarlo para ser sustituido.



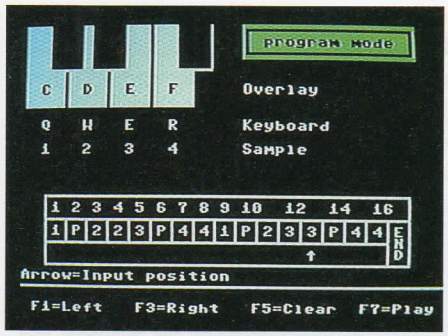
PONIENDOLO A TRABAJAR

Insertar el **Sampler** en el ordenador no presenta diferencia alguna con respecto a cualquier otro cartucho. Hay que tomar igualmente la precaución de desconectar el ordenador para enchufar o desenchufar el dispositivo.

El complemento obligado de este dispositivo es el *software*, que se ofrece en versiones para *cassette* y disco. Nosotros hemos probado esta última.

Cargamos el programa principal del modo habitual: LOAD «*»,8,1. Tras un rato de espera, la pantalla presenta su menú de trabajo. A la izquierda podemos observar una barra vertical que sube y baja como un termómetro, indicando el nivel instatáneo de la señal que llega al **Sampler**.

A partir de aquí, el manejo es sorprendentemente sencillo. Prácticamente con tres de las teclas de función —f1, f3 y f7— se manejan todos los menús que conducen a la opción deseada. La organización de los mismos es jerárquica, visualizándose en forma de ventanas que van superponiéndose a medida que elegimos entre las alternativas posibles.



MENUS PARA TODO

El menú principal ofrece las siguientes opciones:

**Sound Sampler*, muestreador de sonido.

**Quatro Sampling*, manejo de cuatro muestras de sonido.

**Pitch Convert*, alterador de tono fundamental.

**Echo*, cámara de eco.

**Disk utilities* manejo del disco.

**MIDI on/off*, empleo del *interface* MIDI.

**MIDI byte 1.*

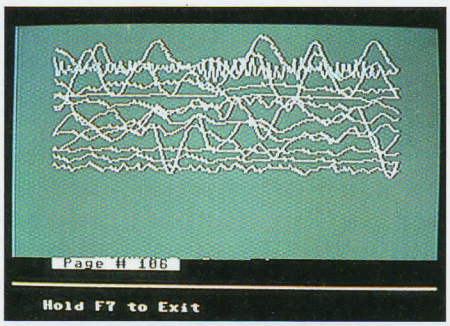
**MIDI channel 1.*

Seguidamente pasamos a describir algunas de estas opciones.

En todas las ventanas aparece una banda de distinto color que el fondo que puede ser desplazada hacia arriba y abajo mediante las teclas f1 y f3. La opción puede elegirse en cada momento mediante f7.

Eligiendo «*Sound Sampling*» el programa visualiza una ventana que propone nuevas leyendas que con la misma filosofía de la banda de color deslizante.

Desde ella se puede digitalizar una muestra de sonido con un duración de hasta 1,4 segundos. Sólamente es necesario pulsar la barra espaciadora para que la señal procedente del microfono, u otra fuente de audio, fluya hacia la memoria del ordenador.

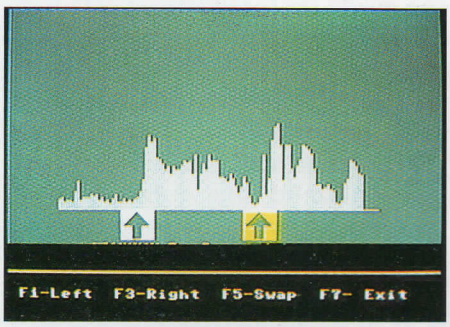


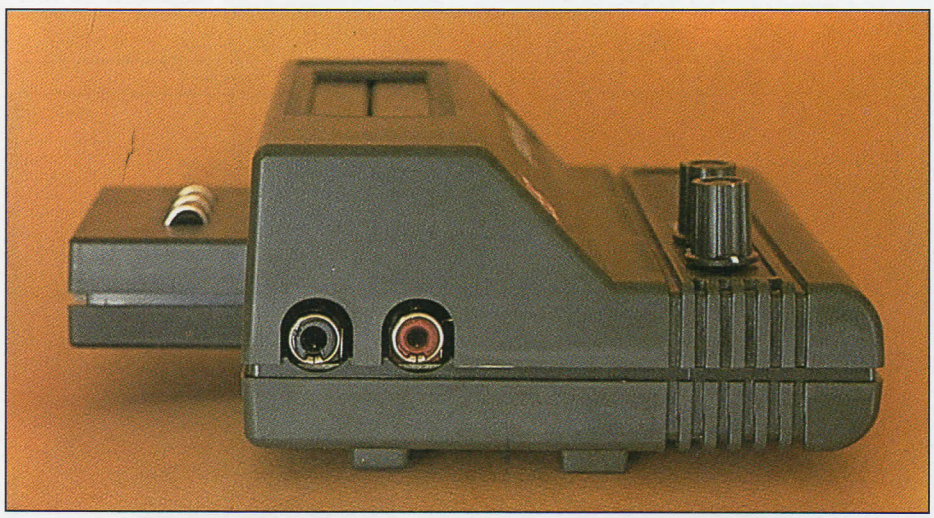
A partir de este momento podemos reproducir ese sonido en modo normal o inverso, de atras hacia adelante. También puede utilizarse el teclado a imagen y semejanza de un piano, donde la escala cromática reproduce la muestra con el tono asociado a cada tecla, de tal forma que se puede llegar a ser virtuoso del chirrido, siempre que sea este el sonido almacenado previamente. Bromas aparte, este puede ser un método tan bueno como otro para experimentar con nuevos sonidos. Como ampliación es posible variar la frecuencia de trabajo en una gama tan extensa como son las diez octavas previstas.

Este menú incluye un diapasón, que puede ser activado para afinar un instrumento convencional y que suene en la misma escala.

La opción «Bucle» tiene que ver con la autorepetición de las teclas. Desde el momento en que se elige, si cualquier tecla es pulsada durante un tiempo mayor que el de la muestra, se repetirá el mismo ciclo indefinidamente hasta que sea levantado el dedo. La zona de la muestra que será repetida puede editarse eligiendo la opción «*Edit*». En la pantalla aparece un osciloGrama del sonido y en su base dos flechas que, desplazadas por el usuario, permiten poner los límites al trozo de muestra adecuado.

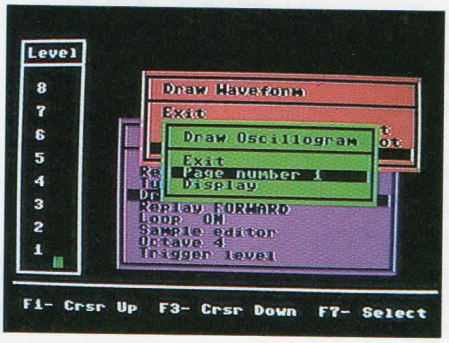
Desde este mismo menú se ajusta el umbral mínimo de sensibilidad del micrófono, de tal manera que es posible, por ejemplo, dejar fuera el ruido ambiente de fondo con sólo situar la flecha en el lugar de la barra de volumen considerado idóneo. Para ello se siguen utilizando las mismas teclas de función del **C-64**.





Para acabar con este menú, veamos otra interesante opción que nos permite observar la conformación global del sonido grabado. Se trata del dibujo de la forma de onda, «*Draw waveform*». Existen tres modos de visualización. Para hacernos una idea más aproximada de su forma de operación, debemos aclarar que la zona de almacenamiento del sonido se divide por comodidad en áreas (páginas) de memoria de 256 bytes cada una.

Podemos ir a la leyenda «*OsciloGram*», que visualiza el diagrama de una de estas páginas en los ejes amplitud/tiempo, como si de un osiclograma real se tratara, avanzando y retrocediendo página a página, con f1 ó f3.



Los oscilogramas de todo el sonido almacenado, comprimidos en ocho páginas por línea, pueden hacerse aparecer siguiendo líneas paralelas, que toman aspecto de montaña. También se puede observar sólamente una zona comprendida entre dos páginas de memoria determinadas.

En el disco hay almacenado un increible sonido de guitarra eléctrica, que puede ser tocada desde el teclado en el mencionado rango de diez octavas.

«*Quatro Sampling*» funciona de un modo parecido al muestreo del apartado anterior, sólo que esta vez la memoria utilizable total se divide en cuatro áreas, en las que se pueden grabar otras tantas muestras de sonido. Después, la tecla Q reproduce la primera, la W la segunda, E la tercera y R la cuarta.

Ahora disponemos de una nueva opción, se trata del secuenciador. Es un sistema que nos permite almacenar una secuencia de hasta dieciséis pasos y cada uno de ellos puede ser uno de los cuatro sonidos o un silencio. El tempo que controla la velocidad de ejecución de la secuencia sonora puede variarse entre muy rápido y muy lento, en un margen dinámico subjetivo de 1 a 28.

El secuenciador puede trabajar igualmente con los sonidos de tambor y palmas existentes en un fichero del *diskette*.

«*Echo*» es un efecto de eco que se produce en tiempo real. Resulta increíble, simplemente eligiendo el tiempo de retardo adecuado puede parecer que estamos hablando en el interior de una majestuosa catedral. Los tiempos son fijos y por pasos, seleccionables entre varios milisegundos y 2 segundos. Situando el micrófono adecuadamente y ajustando los potenciómetros de volumen y realimentación con cuidado, es posible simular la cadena de ecos que produciría el interior de una gruta, hasta que la distancia ahoga nuestra voz.

Los efectos de la siguiente opción son exclusivamente una transformación del tono del sonido aplicado a la entrada de audio. Se ha previsto un total de ocho, que van desde el grave 1 hasta el elevado 8, que hace a nuestra voz paracerse a la del popular **Pato Donald** sin necesidad de hacer nada especial aparte de hablar normalmente por el micrófono. El 1 divide la frecuencia de reproducción por dos, produciéndolo una octava por debajo. El efecto 8 la sube una octava y, por ejemplo, el 2 la baja un quinto.

Las opciones para disco son cuatro clásicas: Formateo de un *diskette*, creación de un fichero de sonido, carga en el ordenador de un fichero ya existente y borrado de uno que deja de interesarnos.

Por último, la opción **MIDI** es de gran valor para los iniciados en la música. A través de este *interface* universal (*Interface* Digital Para Instrumentos Musicales) es posible enlazarse con diversos instrumentos de música electrónica y ordenadores.

Por ejemplo, un teclado **MIDI** podrá conectarse a través de la ranura para cartuchos que tiene el **Sound Sampler**. Está prevista la utilización de los **MIDI** tipo 1 ó 2. También es factible abrir un canal de control si se quiere controlar más de un muestreador desde el exterior.

El **Sound Sampler** ofrece más posibilidades que las de un simple digitalizador de audio, abriendo nuevos horizontes a quienes gusta la experimentación musical y no disponen de muchos recursos.

GANADORES DE LOS MEJORES DE INPUT COMMODORE

En el sorteo correspondiente al número 9 entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

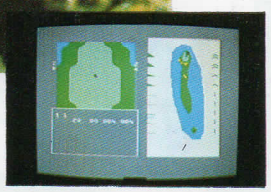
NOMBRE	LOCALIDAD	JUEGO ELEGIDO
F. José Gómez Núñez	Castelldefels (Barcelona)	Revs
José Ignacio Lebrato Rojo	Valladolid	Saboteur
Yolanda Bandrés Ottavi	Barcelona	Uridium
José Andrés Molina Martínez	Archena (Murcia)	Green Beret
Roberto Fernández Palenzuela	Barcelona	Green Beret
José Antonio Sañudo Cano	Tarrasa (Barcelona)	Skyfox
José Luis Serranoi Herreros	Benetuser (Valencia)	Kung Fu Master
Luis Alberto González Gómez	León	Desert Fox
Daniel Fdez. De la Oliva	Madrid	Gi-Joe
David Ruiz Medina	Málaga	Commando

¡JUEGA AL MASTER EN TU CASA...
...Y GANA A LOS MEJORES!

The GOLF
CONSTRUCTION SET

Pantallas e Instrucciones en CASTELLANO

Ya puedes organizar tu propio Torneo de Golf sin salir de casa.
Escucha esto:
Construye tu propio "CAMPO IDEAL", caminos, hoyos, lagos, bancos de arena, pendientes y otras ideas diabólicas.
Utiliza y selecciona tu juego de palos (5 maderas y 9 hierros).



Golpea en el ángulo idoneo, teniendo en cuenta:
- *La dirección del viento.*
- *El estado del cesped (duro o blando).*

Distintos niveles de dificultad. juego suleto o Torneos completos. ¡Y varios circuitos "reales" programados "ya" en tu ordenador para que juegues codo con codo con los mejores!.
¡Esta vez el master puede ser tuyo!.

DRO SOFT

Editado por DRO SOFT. Fundadores, 3 - 28028 Madrid
Tlfs.: 255 45 00 / 09

ariolasoft

PONIENDO LAS COSAS EN ORDEN

- ¿QUE ES LA ORDENACION?
- EL METODO DE LA BURBUJA
- LOS MODULOS DE SHELL Y SHELL-METZNER
- MEJORA DE LA VELOCIDAD

Todo programa que tenga que manejar datos, puede beneficiarse del uso de una rutina de ordenación del tipo que sea. Aquí nos ocuparemos de tres de los métodos de ordenación más comunes.

La ordenación es uno de los aspectos más básicos del proceso de la información. Los ordenadores son muy buenos para la manipulación de datos con gran rapidez y la ordenación juega un papel muy importante al hacer que esta información sea fácilmente accesible para su inspección y corrección.

Imagínate lo difícil que sería seguirle la pista a cualquier voz en particular dentro de un libro si éste no tuviera índice. O consultar un número de teléfono en una guía que no estuviera ordenada alfabéticamente. Ambas son listas esenciales de información que, a diferencia de las entradas aleatorias de algo en una lista de la compra, contienen sus elementos ordenados siguiendo un determinado criterio, en este caso el alfabético. Pero también podrías fijar tu atención en una lista de elementos ordenados numéricamente, como por ejemplo las entradas de cheques en una sucursal bancaria.

Hay muchos tipos de programas en los que se utilizan rutinas de ordenación; también ocurre esto en los juegos en los que se hace la clasificación con arreglo a una puntuación.

Pero es más corriente encontrar rutinas de ordenación en los programas han de manejar la información en listas, tales como listas de direcciones, datos personales, etc.

¿QUE ES LA ORDENACION?

Se llama ordenación o clasificación simplemente a colocar la información en un orden especificado, tal como se hace cuando se abre en abanico un grupo de cartas y se van poniendo por orden de números y palos.

En las ordenaciones de tipo numérico, aparecen sucesivamente valores mayores antes o después que valores más pequeños. En las de tipo alfabético, aparecen letras más próximas al comienzo del alfabeto antes o después que otras letras más lejanas. También puedes basar una ordenación en un criterio alfanumérico, en el que se consideran letras y números, aunque siempre debe prevalecer uno de ambos elementos sobre el otro; normalmente las letras tienen prioridad.

En las ordenaciones numéricas, el valor 1 es el que tiene usualmente la máxima prioridad, mientras que en las alfabéticas es la A el elemento de máxima prioridad. Pero para determinados fines, estos criterios pueden cambiarse, asignando por ejemplo al valor 9 y a la letra Z la máxima prioridad.

ORDENACION POR EL METODO DE LA BURBUJA

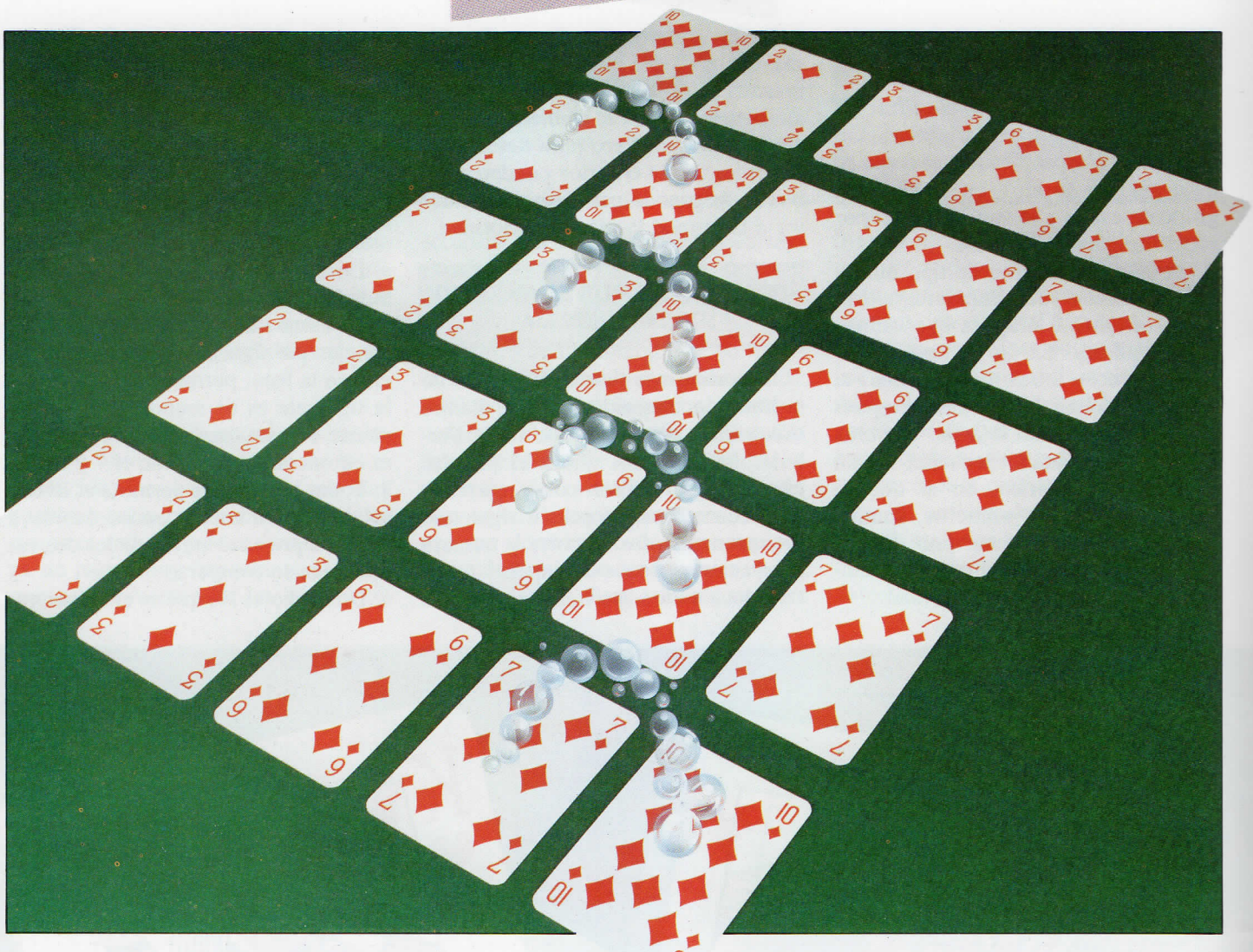
La forma más sencilla de rutina de ordenación conocida se suele llamar ordenación por el método de la burbuja. Examinemos ahora el proceso con detalle. Utiliza como ejercicio unos cuantos naipes para ir siguiendo los razonamientos, o corta y numera unos cuantos trozos de papel.

Coloca esta selección de cartas en una mesa por este orden, con el «tres» en la posición más alejada de tí:

```
ULTIMO          [TRES]
                [SIETE]
                [DOS]
                [SEIS]
PRIMERO         [DIEZ]
```

En el método de ordenación de la burbuja, el primer valor —en este caso la carta número 10— se compara con la que figura a continuación de ella en la lista, permutándolas cuando la siguiente es de menor valor (como ocurre en este caso). El 10 se compara entonces con el 2, con el 7 y con el 3. Todas las veces se permuta el 10 con las que se va encontrando, debido a que siempre es el mayor de los dos valores que se comparan.

De hecho el 10, que es el mayor va-



lor en todas las comparaciones realizadas, ha «burbujeado» a través de los restantes valores, lo que explica el nombre por el que se conoce popularmente a esta rutina de ordenación. Por la misma razón, los valores más pequeños burbujean hacia abajo. Esta ordenación de la burbuja se suele llamar ordenación por cambio u ordenación por intercambio.

Así, después de una pasada, la fila de cartas quedaría de la siguiente forma:

```
ULTIMO      [DIEZ]
            [TRES]
            [SIETE]
            [DOS]
PRIMERO     [SEIS]
```

El proceso comienza de nuevo ahora con la siguiente «primera» carta, que es el 6, el cual se compara con el 2 y se permuta debido a que es mayor. Pero la siguiente comparación se hace con el 7. Ahora, como 7 es mayor, la ordenación continúa con el 7, dejando el 6 en la última posición en la que obtuvo una comparación con ventaja. La carta número 7 se compara primero con el 3 y después con el 10, desplazándose hacia arriba únicamente un lugar, debido a que se encuentra con un valor mayor que el suyo. El nuevo orden de las cartas es ahora el siguiente:

```
ULTIMO      [DIEZ]
            [SIETE]
            [TRES]
            [SEIS]
PRIMERO     [DOS]
```

La nueva primera carta, el 2, se compara ahora con su vecina, el 6. La carta número 6 es la que prevalece ahora por ser mayor, permutándose con la número 3, pero ya no puede llegar más lejos al compararse con la carta número 7. La ordenación llega entonces a su fin al no poder producirse más intercambios, con lo que el orden final de las cartas queda de la siguiente manera:

```
ULTIMO      [DIEZ]
            [SIETE]
            [SEIS]
            [TRES]
PRIMERO     [DOS]
```

Un ordenador realizaría de hecho una última pasada por esta última serie ordenada, para ver si hay algún par más que admita un intercambio, dete-

niéndose únicamente al no poder hacer ninguno más.

En el curso de una ordenación por burbuja, se produce un determinado número de comparaciones e intercambios. Estas dos operaciones diferentes forman parte de toda rutina de ordenación, y la diferencia entre unas rutinas y otras depende enteramente del número de estas operaciones y del tiempo requerido. Veamos un ejemplo:

```
ARRIBA:                ABAJO:
67 35 72 19 47 38 11 96
```

El primero de esta lista (arriba) es el 67. Este valor se compara con 35 y se intercambia con él, pero ya no se mueve más. El valor mayor que interviene en las comparaciones es ahora 72, se compara y se va permutando con 19, 47, 38 y 11 hasta que se encuentra con 96, momento en que la rutina vuelve al principio comenzando con su nuevo primer número. En la tabla 1 se muestra el proceso paso a paso, se presentan encerrados entre paréntesis los elementos que van siendo sometidos a comparación. Si en la fila siguiente aparece un número diferente en esa misma columna, es que se ha producido un intercambio.

¿Cómo sería entonces un programa de ordenación por el método de la burbuja? Ensaya el siguiente programa en el que la rutina de ordenación de la burbuja ocupa la subrutina que empieza en la línea 1000. Almacena el programa, al que podrás añadir después otras rutinas de ordenación.

Teclea para Commodore-64

```
10 PRINT:PRINT:INPUT"NUMERO
   DE COMPONENTES";AA:IF AA<
   2 THEN 10
15 DIM A(AA)
20 PRINT:PRINT"TABLA SIN
   CLASIFICAR":PRINT
30 FOR Z=1 TO AA
40 A(Z)=INT(RND(1)*100)+1
50 PRINT A(Z),
60 NEXT Z
70 PRINT:PRINT:PRINT
   "PRESIONA C PARA
   CLASIFICAR"
80 GET K$:IF K$<>"C" THEN 80
90 GOSUB 1000
100 PRINT:PRINT:PRINT "TABLA
    CLASIFICADA"
110 PRINT:FOR Z=1 TO AA
120 PRINT A(Z),
130 NEXT Z
140 RUN
999 REM**CLASIFICACION POR
    EL METODO DE LA
    BURBUJA**
1000 FOR Z=1 TO AA-1
1010 ZZ=0
1020 FOR Y=1 TO AA-1
1030 IF A(Y+1)<A(Y) THEN X=
     A(Y):A(Y)=A(Y+1):A(Y+1)
     =X:ZZ=1
1040 NEXT Y
1050 IF ZZ=0 THEN RETURN
1060 NEXT Z:RETURN
```

Teclea RUN para ejecutar el programa. Empieza preguntándote cuántos elementos quieres ordenar, creando un conjunto de números pseudoaleatorios basado en tu respuesta. Estos números se presentan en la pantalla bajo el encabezamiento «Tabla sin ordenar». A continuación aparece un mensaje preguntándote si estás dispuesto para empezar la ordenación. Para empezar no tienes más que teclear «C».

Se produce entonces la ordenación. Con unos pocos elementos puede que no tarde más que un segundo o poco más, pero prueba a contestar al mensaje de apertura con el número 50, por ejemplo. Una vez que se haya completado el proceso de ordenación, se presentará el nuevo grupo de números junto a los otros.

TABLA 1

Arriba							Abajo
(67)	(35)	72	19	47	38	11	96
35	(67)	(72)	19	47	38	11	96
35	67	(72)	(19)	47	38	11	96
35	67	19	(72)	(47)	38	11	96
35	67	19	47	(72)	(38)	11	96
35	67	19	47	38	(72)	(11)	96
35	67	19	47	38	11	(72)	(96)
(35)	(67)	19	47	38	11	72	96
35	(67)	(19)	47	38	11	72	96
35	19	(67)	(47)	38	11	72	96
35	19	47	(67)	(38)	11	72	96
35	19	47	38	(67)	(11)	72	96
35	19	47	38	11	(67)	(72)	96
35	19	47	38	11	67	(72)	(96)
(35)	(19)	47	38	11	67	72	96
19	(35)	(47)	38	11	67	72	96
19	35	(47)	(38)	11	67	72	96
19	35	38	(47)	(11)	67	72	96
19	35	38	11	(47)	(67)	72	96

Arriba							Abajo
19	35	38	11	47	(67)	(72)	96
19	35	38	11	47	67	(72)	(96)
(19)	(35)	38	11	47	67	72	96
19	(35)	(38)	11	47	67	72	96
19	35	(38)	(11)	47	67	72	96
19	35	11	(38)	(47)	67	72	96
19	35	11	38	(47)	(67)	72	96
19	35	11	38	47	(67)	(72)	96
19	35	11	38	47	67	(72)	(96)
(19)	(35)	11	38	47	67	72	96
19	(35)	(11)	38	47	67	72	96
19	11	(35)	(38)	47	67	72	96
19	11	35	(38)	(47)	67	72	96
19	11	35	38	(47)	(67)	72	96
19	11	35	38	47	(67)	(72)	96
19	11	35	38	47	67	(72)	(96)
(19)	(11)	35	38	47	67	72	96
11	19	35	38	47	67	72	96

Para poder comparar unos métodos con otros, es útil tomar el tiempo requerido por el proceso de ordenación. El cronometraje es algo que puede hacer muy bien el ordenador por tí. Incorpora esta rutina de temporización dentro del programa existente:

Teclea para C-64 y Vic-20

```
90 TI$="000000":GOSUB 1000:
   PRINT:PRINT"TIEMPO: ";
   TI/60;"SEGUNDOS"
```

Para que empiece el proceso de medida de tiempos, y la rutina de ordenación, pulsa la tecla C cuando aparezca el mensaje de comienzo. El tiempo transcurrido hasta el momento en que se completa la rutina aparece en la pantalla y la ejecución del programa arranca de nuevo automáticamente.

Aunque los tiempos de ordenación son probablemente mucho más rápidos que los que podrías lograr ordenando tú directamente los números, para las magnitudes de tiempo que se manejan con los ordenadores se trata de un tiempo desesperadamente largo. Por esta razón no se utiliza normalmente dicha rutina en este estado tan «crudo» en los programas de proceso de datos, a no ser que se trate de listas muy cortas.

Pero asombrosamente existe un ejemplo en el que la rutina del método de la burbuja proporciona realmente los resultados más rápidos. Se trata del caso en que se utiliza para reordenar una lista ya ordenada en la que se ha agregado un nuevo elemento.

Por ejemplo en un programa de direcciones postales, las nuevas direcciones se ordenan separadamente y después se mezclan con la lista vieja. Pero también se pueden ir añadiendo direcciones de una en una a la lista vieja, sometiendo después todo el lote a una nueva ordenación.

Bajo estas circunstancias, el método de la burbuja no tiene que hacer nada más que comparar la nueva dirección con todos sus vecinos hasta que llegue a su posición correcta. Y esto puede hacerse muy rápidamente, por lo que se utiliza muchas veces con buen resultado dentro de rutinas de ordenación de otro tipo.

ORDENACION DE SHELL

La llamada ordenación de **Shell** es una de las dos alternativas más conocidas al método de la burbuja, las cuales han de ser tomadas en consideración cuando el número de elementos a ordenar pasa de unos diez. (Date cuenta que la comparación de una nueva entrada única con una lista ya ordenada, significa que hay que comparar únicamente dos elementos).

La ordenación de **Shell** utiliza una técnica de búsqueda binaria que opera dividiendo sucesivamente en mitades la lista original no ordenada hasta que puede establecerse la posición relativa del nuevo elemento introducido. En cada ocasión interviene en la comparación un nuevo par de valores hasta que se hace una nueva pasada sin permutaciones.

Un módulo de ordenación de **Shell** tiene la forma típica que se muestra a continuación. Añádele esta subrutina a tu programa anterior, modificando el GOSUB que hay en la mitad de la línea 90 para probarla:

Teclea para C-64 y Vic-20

```
1999 REM**CLASIFICACION POR
     EL METODO DE SHELL
2000 Z=AA
2010 IF Z<=1 THEN RETURN
2020 Z=INT(Z/2):Y=AA-Z
2030 ZZ=0
2040 FOR X=1 TO Y
2050 XX=X+Z
2060 IF A(X)>A(XX) THEN YY=
     A(X):A(X)=A(XX):A(XX)=
     YY:ZZ=1
2070 NEXT X
2080 IF ZZ>0 THEN 2030
2090 GOTO 2010
```

¿Cómo funciona este programa? La forma más sencilla de ver lo que sucede es analizar un ejemplo en detalle. Imagínate que se te presenta la siguiente lista de números para su ordenación:

```
ARRIBA:              ABAJO:
67 35 72 19 47 38 11 96
```

En la ordenación de **Shell**, se divide la lista inicialmente en dos partes y se comparan los valores del primer elemento de cada uno de los grupos. Cuando el valor más cercano al principio de la lista es mayor, se efectúa un intercambio. Al dividir en dos grupos se obtienen lo valores iniciales de 67 y 47 que se comparan y se intercambian:

```
ARRIBA:              ABAJO:
47 35 72 19 : 67 38 11 96
```

Después de un cambio, se compara el siguiente par de valores, que en es-

te caso son 35 y 38. En este caso no se produce intercambio. A continuación se comparan 72 y 11, permutándose. Por último se comparan 19 y 96. La lista queda así:

```
ARRIBA:                ABAJO:
47 35 11 19 : 67 38 72 96
```

En este punto cada una de las mitades se subdivide de nuevo:

```
ARRIBA:                ABAJO:
47 35:11 19:67 38:72 96
```

El primer valor, que es 37, se compara con el valor que figura en tercer lugar, el 11, intercambiándose con él. El 35, que figura en segundo lugar, se compara con el 19 que está en cuarto lugar, realizándose una nueva permutación. El 35 queda ahora en la cuarta posición. El valor 47, que se ha sometido con éxito a una comparación, se compara con 67, pero esta vez sin éxito. El cuarto valor, que ahora es 35, se compara con 38 quedándose donde estaba. El quinto valor, 67, se compara con el de la séptima posición, que es el 72, quedando ambos donde estaban. El 38, que figura en sexta posición, se compara sin éxito con 96. En la tabla 2 se muestra la secuencia. Como ocurría antes, se muestran entre paréntesis los valores que intervienen en la comparación: para saber si se ha

TABLA 2

```
[47]  35 :[11]  19 : 67   38 : 72   96
 11  [35]: 47  [19]: 67   38 : 72   96
 11   19 :[47]  35 :[67]  38 : 72   96
 11   19 : 47  [35]: 67  [38]: 72   96
 11   19 : 47   35 :[67]  38 :[72]  96
 11   19 : 47   35 : 67  [38]: 72  [96]
 11   19 : 47   35 : 67   38 : 72   96
```

La velocidad de todas las rutinas de ordenación puede aumentarse considerablemente si la información se presenta ya de alguna forma parcialmente ordenada. Esto es especialmente cierto en el caso sencillo de la ordenación por el método de la burbuja.

Por ejemplo, si los elementos tienen que ser ordenados por orden de fechas, será muy útil introducir los datos en la forma Año/Mes/Día en vez de la forma más usual Día/Mes/Año. La rutina podría ocuparse de los años con mucha rapidez, para generar una primera lista ordenada por años. Esta lista puede ordenarse ahora por meses dentro de cada año. Finalmente, se puede poner orden dentro de cada mes en la parte que requiere más permutaciones, que es la de los días.

Utilizando este método, la ordenación se realiza primeramente dentro de todo el grupo y a continuación en partes separadas. En cualquiera de las fases, los elementos están en cierta medida ordenados.

El proceso anterior es más rápido que llamar a una rutina que vaya examinando y ordenando todos los días, después reordene algunos de estos días debido a la influencia de los meses y por último realice una tercera ordenación debida a la influencia de los años.

Se podría utilizar una subrutina adicional para convertir una secuencia de entrada «normal» en una secuencia de fechas «invertida» para llevar a cabo la ordenación

```
TABLA 3     [11]:[19]: 47 : 35 : 67 : 38 : 72 : 96
             11 :[19]:[47]: 35 : 67 : 38 : 72 : 96
             11 : 19 :[47]:[35]: 67 : 38 : 72 : 96
             11 : 19 : 35 :[47]:[67]: 38 : 72 : 96
             11 : 19 : 35 : 47 :[67]:[38]: 72 : 96
             11 : 19 : 35 : 47 : 38 :[67]:[72]: 96
```

producido o no un cambio, tienes que observar lo que ocurre en la línea siguiente.

La lista sufre una nueva división para dar los siguientes grupos:

```
ARRIBA:              ABAJO:
11:19:47:35:67:38:72:96
```

Los valores se comparan de nuevo, abriéndose camino los valores más bajos hacia el principio de la lista. Observa que la rutina solamente compara bloques adyacentes. El primero se compara con el segundo, el segundo con el tercero, y así sucesivamente a lo largo de toda la lista, como se observa en la tabla 3.

La ordenación continúa hasta llegar a los valores finales que ya están en orden, volviendo de nuevo al principio, como se observa en la tabla 4.

Aunque la lista ya está ordenada, el programa realizará una nueva pasada. Cualquier valor nuevo puede entonces ser comparado con esta lista ordenada de la misma forma que en el método de la burbuja, ya que eso es esencialmente lo que se hace en cada comparación binaria.

ORDENACION DE SHELL-METZNER

La ordenación por el método de **Shell-Metzner** utiliza una rutina de ordenación binaria aún más rápida, que se deriva del algoritmo de **Shell**:

Teclea para C-64 y Vic-20

```
2999 REM**METODO DE
     SHELL-METZNER
3000 Z=AA
3010 Z=INT(Z/2)
3020 IF Z=0 THEN RETURN
3030 Y=AA-Z:ZZ=1
3040 X=ZZ
3050 XX=X+Z
3060 IF A(X)<=A(XX)THEN 3090
3070 W=A(X):A(X)=A(XX):A(XX)
     =W:X=X-Z
3080 IF X>=1 THEN 3050
3090 ZZ=ZZ+1
3100 IF ZZ<=Y THEN 3040
3110 GOTO 3010
```

Añade esta rutina a tu anterior programa de demostración, modificando la instrucción GOSUB de la línea 90 y pruébala.

Como puedes ver, en comparación con el método de la burbuja es extremadamente rápida para el caso en que tengas cincuenta números aleatorios. Pero haz una prueba con conjuntos de números realmente grandes. Mientras trabaja el método de la burbuja puedes irte a tomar un café o a hacer cualquier otra cosa.

```
TABLA 4     [11]:[19]: 35 : 47 : 38 : 67 : 72 : 96
             11 :[19]:[35]: 47 : 38 : 67 : 72 : 96
             11 : 19 :[35]:[47]: 38 : 67 : 72 : 96
             11 : 19 : 35 :[47]:[38]: 67 : 72 : 96
             11 : 19 : 35 : 38 :[47]:[67]: 72 : 96
```

RESERVA TU INPUT ESPECIAL VERANO

Que sorteará más de **100 REGALOS**

AMIGA by Commodore

TARDAREMOS EN VER ALGO PARECIDO.

En informática ya es difícil sorprender. Pues bien, Commodore lo ha conseguido con Amiga.

Más que una nueva generación de ordenadores, el Amiga de Commodore representa un nuevo concepto. Un ordenador que ofrece una serie de posibilidades y abre unos caminos que hasta hoy eran impensables.

Entre las novedades de este ordenador profesional destacan: la posibilidad de trabajar con un procesador de 32 bits y de 3 coprocesadores específicos, actuando los 4 al tiempo.

Además, Amiga dispone de pantallas y ventanas configurables con una resolución de 640 x 400 puntos escogiendo entre una paleta de 4.096 colores.

Trabaja con 4 canales independientes polifónicos con voz masculina y femenina, y por si fuera poco, Amiga de Commodore es el único ordenador multitarea que puede efectuar varios trabajos simultáneamente.

El precio también es importante. Por 330.000 ptas. ex. IVA, el Amiga incluye:

- Unidad central con 768 K RAM.
- Monitor de alta resolución en color y sonido.
- Unidad de discos de 880 K.
- Teclado profesional y ratón.

Por mucho que avance la investigación en el campo de los ordenadores, tardaremos en ver algo parecido.

commodore

Imagina siempre lo mejor.

Microelectrónica y Control, S.A. *Unico representante de Commodore en España.*
c/ Valencia, 49-53 Tel. (91) 246 37 36 08015 Barcelona - c/ Ardemans, 8 Tel.(93) 325 57 02 28028 Madrid

¿QUIEN ASESINO A LOS JUEGOS?

Un misterioso cartucho, capaz de evitar que seamos eliminados o eliminemos a alguien en un buen puñado de juegos comerciales, se está poniendo de moda en Gran Bretaña.

En apariencia nada lo distingue de cualquier otro cartucho conectable al *port* del usuario del **Commodore 64**. Sin embargo tiene poderes casi mágicos, es capaz de inhibir las interrupciones producidas por el choque entre *sprites* o entre *sprite* y fondo, consiguiendo así que el programa no se entere de que el oponente ha logrado alcanzarnos, aunque también puede ocurrir lo contrario: que se eliminen nuestras posibilidades de cazar.

Si bien en principio puede resultar divertido, también puede convertirse en la réplica del clásico amiguete capaz de asesinar un chiste contando el final apenas alguien comienza a contarlo, porque se los conoce todos.

Como todos los cartuchos, se enchufa con la alimentación del ordenador desconectada. Acto seguido se pone el ordenador en marcha y comienza a sonar una agradable melodía al tiempo que una pantalla de presentación nos hace ver lo que es un buen *scroolling* combinado con el baile de *sprites* en forma de letras. Cuando nos cansamos de verlo, no hay mas que presionar la barra espaciadora y la pantalla vuelve a ser la misma que aparece siempre que conectamos el ordenador por vez primera.

Llegados a este punto, ya podemos cargar el programa que más nos guste.

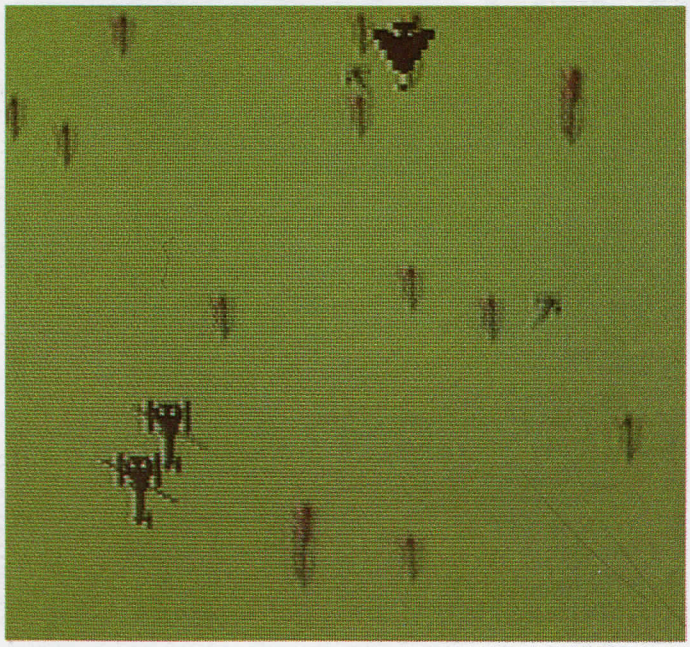
Cansados de ser muertos a la primera de cambio, podemos ver si el **Games Killer** tiene algo que hacer con nuestro juego. Para ello disponemos de la ayuda del botoncito rojo, que asoma por el lateral de cartucho. Al presionarlo desaparece momentáneamente la imagen de la pantalla y vuelve de inmediato al mismo punto.

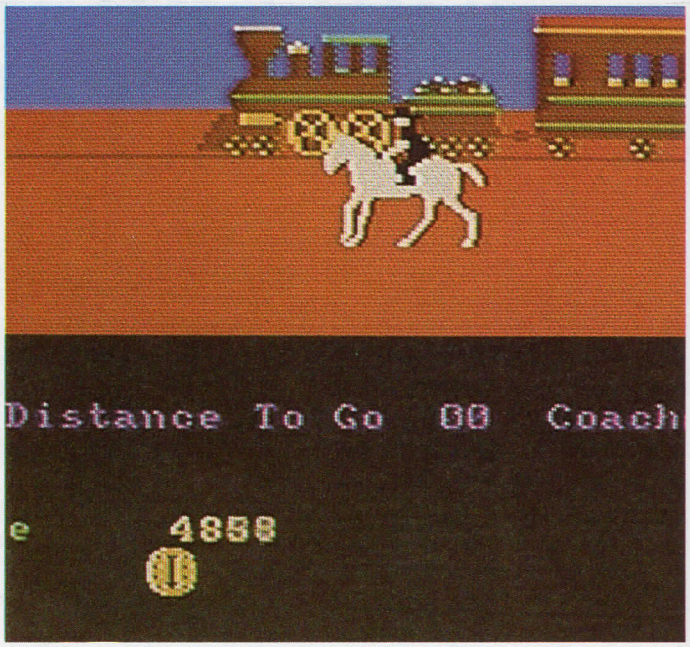
Como decíamos, podemos optar por la desactivación de las colisiones entre *sprites*, entre *sprites* y fondo o por ambas. Este último modo se consigue presionando sólo el botoncito.

Si previamente hemos empujado el *joystick* hacia arriba y, manteniéndolo, activamos el botón obtenemos una de las inhibiciones mencionadas. Empujándolo hacia abajo (y presionando en el cartucho) conseguimos la otra inhibición. La combinación con el botón de disparo del *joystick* reactiva las colisiones.

Lo hemos comprobado con varios juegos. Recordamos lo difícil que resultó hace un par de años jugar las primeras partidas con **Tales of the Arabian Nights**.

Bien, ahora es un juego de niños con el cartuchito. No te matan las caidas, ni los pulpos y los balazos, ni los cocodrilos, ni los fantasmas... Juegos de gran popularidad actualmente, como **Kane** permiten que te vengues de sus dificultades: Los patos no caen cuando les atraviesa la flecha, el caballo pasa tras las rocas y los arbustos sin problemas, etc.





Ejemplos de indestructibilidad

Con **Mission Impossible** puedes disfrutar como un mandarín chino, no hay robot que te fría el cuerpo.

Sin embargo, en **Uridium** no hay *niente da fare*, los obstáculos no sienten piedad por ti. En **Rock'n'Lucha** ocurre tres cuartos de lo mismo.

Este cartucho desde luego no es la panacea universal, puesto que en gran cantidad de juegos es sencillamente inaplicable. Pero es una valiosa herramienta para saber lo que ocurriría si...

HACER SALTAR A LA BANCA

- EL ORDENADOR Y EL JUEGO DE LAS VEINTIUNA
- JUGANDO CON EL ORDENADOR
- COMO FUNCIONA EL PROGRAMA
- LAS PIEDRAS

Para completar el juego de las veintiuna, tienes que programar tu ordenador para que juegue cuando le toque el turno. Aquí tienes además una descripción de las reglas del juego para el caso de que no estés familiarizado con el mismo.

Antes de pasar a examinar el resto de la programación, vale la pena recapitular las reglas del juego.

El juego de las veintiuna se juega con un mazo ordinario de 52 cartas. Las cartas del 2 al 10 valen lo que indica su número, las figuras valen 10 y el as vale 1 u 11, según las necesidades del jugador. En este juego el ordenador se ocupa de ir llevando la cuenta de los totales, por lo que no tienes que hacerlo tú.

Normalmente se juega con dinero, o con «piedras», pero en esta versión para ordenador hay que programar el mismo para que cuente la puntuación en piedras imaginarias; con las otras piedras un programador poco escrupuloso podría hacer que la máquina dejara de ser honrada. El ordenador se programa para actuar como banquero en todo momento y será siempre el encargado de repartir las cartas. A las piedras les llamaremos *chips* en el programa.

Al principio del juego se barajan las cartas y se sacan dos de ellas, poniéndolas boca abajo.En la pantalla aparecerá boca arriba la carta del jugador, aunque el programa ha sido diseñado para que la máquina no sepa qué carta es la que tiene el jugador.

El jugador debe apostar ahora sobre esta primera carta, antes de que se saque una nueva para cada uno.

El objeto del juego es terminar con mejor puntuación que la banca, es decir con un valor total más elevado. Una mano que sume en total más de 21, se «pasa» y pierde. Una mano con puntuación entre 16 y 21 solamente vence a la banca si la máquina tiene una mano más baja o se ha pasado de

21. Sólo hay dos triunfos especiales: las veintiuna, consistentes en un as y un diez o una figura, que en total suman 21 con sólo dos cartas, y las cinco cartas, que es cualquier grupo de cinco cartas con un total de 21 o menos. Las veintiuna del jugador vencen a todo lo que tenga la banca excepto a otras veintiuna. Las cinco cartas también vencen a todo lo que tenga la banca excepto a las veintiuna y a otras cinco cartas.

FUNCIONAMIENTO DEL PROGRAMA

Después de que el jugador ha recibido una segunda carta, el programa comprueba si se han hecho veintiuna. De no ser así, al jugador se le ofrecen tres opciones. En primer lugar, si totaliza con sus cartas una puntuación de 16 o mayor, y está satisfecho con sus cartas, puede decidir «plantarse». Aquí termina su turno y el juego continúa por parte de la banca. En segundo lugar, si el total está por debajo de 16 o el jugador no está satisfecho con el total, puede adquirir más cartas «robando» o «doblando». En el juego normal de las veintiuna pedir una carta impone al jugador la obligación de hacer una puesta igual a la original. En los cambios el banquero utiliza la carta que hay boca abajo. En la versión para ordenador no se hace distinción entre las cartas boca arriba y boca abajo, por lo que cada carta aparece al lado de la anterior. El robo no implica que el jugador tenga que hacer una nueva puesta, y en un juego real la carta se colocaría boca arriba; aquí no hay diferencia. El jugador no puede doblar después de robar y si está intentando conseguir la última carta para completar un triunfo de cinco cartas, tiene que robar obligatoriamente. Después de extraer cada carta el programa comprueba que el jugador ni la banca han superado los 21 puntos y por tanto no se han pasado.

Si el jugador se pasa, la banca toma toda su puesta y el ordenador ya no tiene que ocuparse de su juego. En caso contrario el banquero da la vuelta a las cartas y tiene que decidir si plantarse o tomar más cartas. Si logra las veintiuna en esta fase, gana automáticamente y ya no se sacan más cartas. Si no tiene las veintiuna, la banca seguirá sacando nuevas cartas hasta que quede satisfecha con el total, con-

siga un triunfo de cinco cartas o se salga.

Si la banca se planta con un numero inferior a 21 y no ha logrado un triunfo de cinco cartas, aparece en la pantalla un mensaje y un número igual al tanteo de las cartas actuales más uno. Si el tanteo es 21, aparece el mensaje SOLO VEINTIUNA O JUEGO DE CINCO CARTAS. Si la banca consigue las cinco cartas, el jugador recibe el mensaje SOLO LAS VEINTIUNA. En un juego ordinario, los jugadores tendrían que declarar a la banca sus cartas en caso de que ganaran, pero en este juego, naturalmente, el ordenador ya tiene la puntuación del jugador.

Si el jugador consigue reunir una puntuación igual a la anunciada, gana, pero si tiene menos pierde su puesta.

Cuando el jugador gana, la banca pone una cantidad igual a la apostada por el jugador, por lo que éste recibe el doble de la cantidad que se jugó. A diferencia de lo que ocurre con las reglas de otros juegos, las veintiuna y las cinco cartas no se pagan con más *chips*.

Otra diferencia con respecto a los juegos ordinarios de cartas, es que el reparto no puede pasar al jugador, debiendo ser siempre el ordenador el que «da» las cartas. En el juego ordinario la banca cambia cuando se logran las veintiuna, en este juego la banca es siempre el ordenador, por lo que siempre tiene ventaja sobre el jugador; en las veintiuna ordinarias, la banca tiene también ventaja sobre el jugador. Por ello el jugador tiene ante sí la difícil tarea de hacer saltar a la banca para ganar, lo que aquí significa acumular más de 1000 *chips*. El jugador empieza a jugar con 100 *chips* y no debe perderlas todas, ya que de ocurrir así perdería el juego completo.

A continuación, añade las siguientes líneas a tu programa de las veintiuna. Están relacionadas principalmente con el turno de la banca:

Teclea para C-64 y Vic-20

```
390 IF PL=2 AND (TU=21 OR(TU
    +T3=21 AND CT(1)=0))
    THEN 750
410 IF PL=2 THEN 710
710 FOR DE=1 TO 500: NEXT DE
720 IF NU=2 AND T(1)>21 THEN
    800
730 IF RND(1)<(21-TU)/13 OR
    TU<16 OR CT(1)=1 THEN
    510
740 IF T3<>0 AND RND(1)<(21-
    (TU+T3))/13 THEN 510
750 PRINT "[CLR/HOME][CRSR
    abajo][CTRL+1]ME PLANTO.
    ..":GOTO 800
780 IF PL=2 THEN S1$="O YO":
    S2$="MI"
800 T(PL)=TU:IF TU+T3<22 AND
    TU+T3>TU THEN T(PL)=TU+
    T3
810 IF PL=1 OR (PL=2 AND CT
    (2)=1) THEN 840
820 IF T(2)<21 THEN PRINT
    "[CLR/HOME][2*CRSR abajo]
    [CTRL+5]PIDE LA BANCA";
    T(2)+1
830 IF T(2)=21 AND PO(2)=0
    THEN PRINT "[CLR/HOME]
    [2*CRSR abajo][CTRL+5]
    SOLO PONTONES O CINCO
    CARTAS!"
840 FOR DE=1 TO 1000*PL:NEXT
    DE:IF PL=1 AND T(1)>21
    THEN 860
850 NEXT PL
860 CH$="CHIPS":IF TB=1 THEN
    CH$="CHIP"
870 IF CT(2)=1 AND (PO(1)=0
    OR CT(1)=1) THEN PRINT
    "[SHIFT+CLR/HOME][CTRL+6]
    PIERDES[CTRL+3]"TB;CH$:
    GOTO 900
875 IF (CT(1)=1 OR PO(1)=1)
    AND PO(2)=0 THEN 890
880 IF(T(2)<22 AND T(2)=>
    T(1))ORT(1)>21THENPRINT
    "[SHIFT+CLR/HOME][CTRL+6]
    PIERDES[CTRL+3]"TB;CH$:
    GOTO 900
890 PRINT"[SHIFT+CLR/HOME]
    [CTRL+6]GANAS[CTRL+3]"
    TB*2;CH$:MU=MU+TB*2
900 IF MU<1 OR MU>999 THEN
    PRINT"[CTRL+6]Y EL JUEGO
    !":GOTO 930
910 PRINT "[CTRL+9][CTRL+7]"
    E2$:FOR DE=1 TO 3000:
    NEXT:IF PO(1)=1 OR PO(2)
    =1 THEN 60
920 GOTO 760
930 PRINT"[CTRL+5][3*CRSR
    abajo]OTRO JUEGO.(S/N)?"
940 GET A$:IF A$="S" THEN
    RUN
950 IF A$<>"N" THEN 940
960 PRINT "[SHIFT+CLR/HOME]"
    :END
```

PL2 es el jugador 2, que tiene la banca. Si la banca tiene 21, decidirá plantarse. En tal caso, la línea 390 comprueba si el ordenador tiene 21, y si es así salta a la línea 350 que anuncia ME PLANTO. La línea 410 hace que el ordenador omita las líneas del turno del jugador.

En la línea 710 comienza el turno propio del ordenador. Hay una pausa antes de una serie de comprobaciones.

La línea 720 comprueba si hay dos ases, mientras que la 730 decide si el ordenador desea tomar otra carta. Si el total es menor que 16, o si el jugador ha acumulado ya un triunfo de cinco cartas, la máquina debe coger otra carta.

PLANTARSE O NO PLANTARSE

La primera parte de dicha línea, la que utiliza la expresión matemática aleatoria, es muy interesante. Está ahí para hacer que el ordenador se comporte de una forma parecida a un jugador humano. La manera más fácil de hacer que juegue el ordenador sería que se plantara siempre con el mismo número, por ejemplo 19. Pero surge el problema de que el jugador se dará cuenta enseguida de cómo debe actuar para batir a la máquina, y siempre intentará superar los 19. Incorporando la aleatoriedad en el bucle IF ... THEN, el comportamiento del ordenador resulta impredecible, pero al mismo tiempo hay que impedir que cambie tan a menudo hacia puntuaciones altas que se pase con demasiada frecuencia. Restando TU de 21, dividiendo por 13 y comparando entonces con el número aleatorio se consigue que la máquina quede más veces entre 16 y 20.

HA SALIDO UN AS

La línea 740 cumple una función parecida a la de la línea 730, pero corresponde al caso en que se ha repartido un as. Si ninguna de las líneas hace que la máquina reciba una nueva carta, la línea 750 informa al jugador de que la máquina se ha plantado, y salta a la línea 800.

En la línea 780 se cambian S1$ y S2$ de forma que sean correctos para la máquina; recuerda que ésta es el jugador número 2.

Cuando ambos jugadores se han plantado, la línea 800 se asegura de que si ha salido algún as, la puntuación con la que se ha detenido el juego es la más alta de las dos alternativas. Si le toca el turno al jugador o el ordenador tiene un triunfo de cinco cartas, el programa utiliza la línea 860 para poner *CHIPS* o *CHIP*. En la línea 850 se cierra el bucle FOR ...NEXT que iniciaste en la línea 760 en la parte anterior de nuestro coleccionable.

GANAR O PERDER

Si el jugador ha perdido, la línea 870 le informa de cuántos *chips* ha perdido. Si por el contrario gana, la línea 875 hace que el programa salte a la línea 890 que informa al jugador de que ha ganado y de cuántos *chips* le corresponden. Finalmente la línea 880 es la otra condición por la que el jugador puede perder. Si el jugador gana o pierde, la línea 900 comprueba si el número de *chips* ha caído por debajo de 1 o ha subido por encima de 1000, en cuyo caso anuncia FIN DEL JUEGO. Seguidamente la línea 930 pregunta si se quiere jugar otra vez; las líneas 930 a 960 son precisamente una rutina de «¿Otra vez?».

Si el jugador o la máquina han conseguido hacer las veintiuna, la línea 910 asegura que las cartas se barajen adecuadamente para la siguiente mano. Si nadie ha hecho veintiuna, el programa no baraja las cartas, y volviendo a la línea 760 se extraen las dos de encima.

CONVIERTE TU AVENTURA EN UNA EPOPEYA

- COMO SE ALMACENA EL TEXTO
- FORMAS DE AHORRAR ESPACIO
- SUSTITUCIONES Y CODIGOS ASCII
- LISTADO DE ENSAMBLADOR

Si eres muy hábil con los juegos de aventuras, pero te sientes frustrado por las limitaciones de memoria de tu máquina, ¿qué te parecería disponer de un programa que permitiera acomodar la misma cantidad de texto en mucho menos espacio?

El mayor problema con el que te encontrarás al escribir programas de juegos de aventuras es que nunca tienes suficiente espacio de memoria para albergar tu última obra maestra. Aparte de comprarte una máquina nueva o una ampliación de memoria, es muy poco lo que puedes hacer, como no sea achicar los horizontes de tu mundo de aventura o simplificar el programa.

La única alternativa que tienes para disminuir la presión ejercida por la escasez de memoria es intentar encontrar la manera de que el programa ocupe menos espacio. Los métodos habituales para acortar programas, no te serán de mucha utilidad, ya que casi todo el programa es texto. Lo que necesitas es precisamente un método que te permita almacenar la mayor cantidad de texto en el menor espacio posible.

En los ordenadores el texto se almacena normalmente en forma de código ASCII. El uso del código ASCII hace que cada carácter ocupe ocho bits, es decir un byte de espacio de memoria.

Si se pudiera apretar cada carácter de forma que ocupe menos de ocho bits, se ahorraría espacio de memoria. Para conseguirlo dispones de varios métodos, cada uno con sus ventajas y sus inconvenientes.

Probablemente la manera más fácil de comprimir el texto es utilizar sólo una parte de la codificación ASCII. Si te conformas con un reducido número de caracteres, por ejemplo sólo mayúsculas, números y algún signo de puntuación, resulta muy fácil almacenar cada carácter en seis bits.

Por ejemplo, si eliges el conjunto de caracteres ASCII que va desde 20 a 5F en hexadecimal, tendrás un buen juego de caracteres para utilizar dentro de tu programa. Si restas 20 hexadecimal del valor ASCII de cada uno de los códigos, el rango quedará reducido a 0 - 3F hexadecimal, que puede almacenarse con sólo seis bits.

El uso de este tipo de codificación te permitirá reducir las exigencias de memoria en una cuarta parte, lo que significa que podrás colocar cuatro caracteres en el mismo espacio en que anteriormente colocabas tres. Para decodificar los caracteres almacenados, no tienes más que ir tomando los bits de seis en seis y sumar 20 hexadecimal al número resultante para obtener el valor ASCII del carácter original.

El mayor inconveniente de una codificación de este tipo es que te ves obligado a elegir un margen de 40 caracteres (en hexadecimal) antes de empezar a programar, debiendo respetarlo estrictamente. Como te encuentres que te hace falta un carácter que no pertenezca al grupo seleccionado, lo vas a tener bastante difícil. Pudiera ocurrir que utilizar únicamente letras mayúsculas resulte inaceptable para los usuarios de máquinas que normalmente hacen aparecer en su pantalla letras minúsculas, por lo que deberás encontrar otra forma de disponer las cosas. En resumen, tienes que buscar otra alternativa.

Existe una posibilidad que se parece a la forma en que opera la lengua china, consistente en asignar un único carácter o valor numérico a cada palabra. Este tipo de codificación te impone la necesidad de decidir el vocabulario completo de tu aventura, asignando a cada palabra un número. La codificación consiste en comparar cada palabra con una lista de datos y almacenar cada código en la memoria. La decodificación es la operación inversa.

Con la «solución china» es posible lograr un uso muy eficiente de la memoria, pero tiene el inconveniente de que hay que volver a escribirla para cada nueva aventura. Naturalmente, siempre puedes empezar con un vocabulario muy extenso y albergar la esperanza de haber elegido las palabras suficientes para tus necesidades, pero el precio de esto es una gran cantidad de memoria. Podrías llegar a encontrarte utilizando números muy grandes como códigos, y la exploración de las listas resultaría muy lenta.

La mejor solución de todas sería algún sistema de compresión de texto que pudiera valer para cualquier aventura; de hecho tendría que valer para cualquier texto que pudieras encontrarte. Dicho «compresor» debería hacer un uso lo más eficiente posible de la memoria disponible, en función de la manera de comprimir el texto y de la cantidad de espacio disponible por el propio *software* de compresión.

Es muy importante que el compresor de texto sea capaz de reconocer un conjunto completo de letras mayúsculas y minúsculas, números y signos de puntuación.

Un esquema de este tipo requiere un planteamiento más radical que los utilizados anteriormente, pero resulta perfectamente posible cumplir todas las condiciones y el fundamento del programa que figura a continuación es precisamente un sistema de este tipo.

SISTEMA DE COMPRESION

Este sistema compresor de textos utiliza algunos principios tomados del mundo de la criptografía, disciplina que se ocupa del estudio de los códigos y los mensajes cifrados. Dichos

principios nos pueden permitir saber algo acerca de la estructura del texto que se somete a compresión.

Cuando se intenta romper un código, es muy útil conocer la frecuencia con que aparecen las distintas letras en el idioma español, ya que ello permite que el descifrador identifique ciertos grupos que intervienen en los mensajes cifrados. Durante mucho tiempo los criptógrafos han trabajado afanosamente sumergiéndose en largos textos y contando el número de veces que se presenta cada letra, cada par de letras o incluso cada palabra.

El compresor de texto que veremos en INPUT está basado en la frecuencia de aparición de las letras sueltas y, en determinados casos, de pares de letras. Se asigna números en binario a cada una de las letras del alfabeto; a las letras más frecuentes se les asignan los números más bajos para ahorrar y a las menos frecuentes se les asignan los números binarios más altos.

Una consecuencia importante de utilizar este tipo de codificación es que, al realizar la prueba del *software* de compresión con grupos aleatorios de letras, sin significado alguno, es muy posible que resulte poca o ninguna compresión. Si por el contrario empleas un texto en español te encontrarás que la compresión es bastante grande. También es interesante observar que si realizas por ejemplo la compresión de un texto en inglés o en francés, el efecto de compresión obtenido es considerablemente menor, debido a que las frecuencias de aparición de cada letra son diferentes en cada idioma, por lo que se requeriría un esquema de codificación diferente.

Además de los códigos de las letras individuales, el compresor también se ocupa de las dos letras más frecuentes que pueden seguir a cada letra. En la tabla 1 tienes un resumen de los códigos del idioma inglés, en el que fué originalmente desarrollado este programa. Más adelante mostraremos como adaptarlo al juego de palabras empleado en tu juego particular o en el idioma español por ejemplo. Observarás que por ejemplo, la letra que sigue a la T con mayor frecuencia es la H, y a continuación la I. Para estas combinaciones de números muy comunes, también se han adoptado por razones de economía códigos especiales.

Cuando el programa compresor está realizando la codificación de un texto, recuerda cuál era la última letra y comprueba si la letra siguiente es la primera o la segunda más frecuentes. Si se trata de una de dichas letras más frecuentes, utiliza el código de la parte superior de la columna. Si por el contrario la letra siguiente no es una de las más comunes, se utiliza el código ordinario de la letra que se presenta sola. Este principio se aplica durante todo el proceso de codificación. Para cada nueva letra o carácter, lo primero que hace la máquina es comprobar si se trata de uno de los dos caracteres más comunes que siguen al anterior.

ACCESO AL COMPRESOR DE TEXTOS

El compresor de textos ha sido escrito en código máquina para acelerar la codificación y decodificación del material a procesar.

El listado se presenta tanto en forma hexadecimal como en lenguaje ensamblador. En esta parte de nuestro coleccionable te presentamos un listado en lenguaje ensamblador correspondiente a la parte del programa encargada de la codificación. En el próximo capítulo del coleccionable veremos la parte de decodificación del listado en ensamblador, así como el listado en forma hexadecimal que completan el programa.

Necesitarás un ensamblador comercial para utilizarlo con el listado de ensamblador, o en caso contrario tendrás que esperar a tener el listado de código máquina. Nosotros hemos empleado el cartucho **Hesmon**.

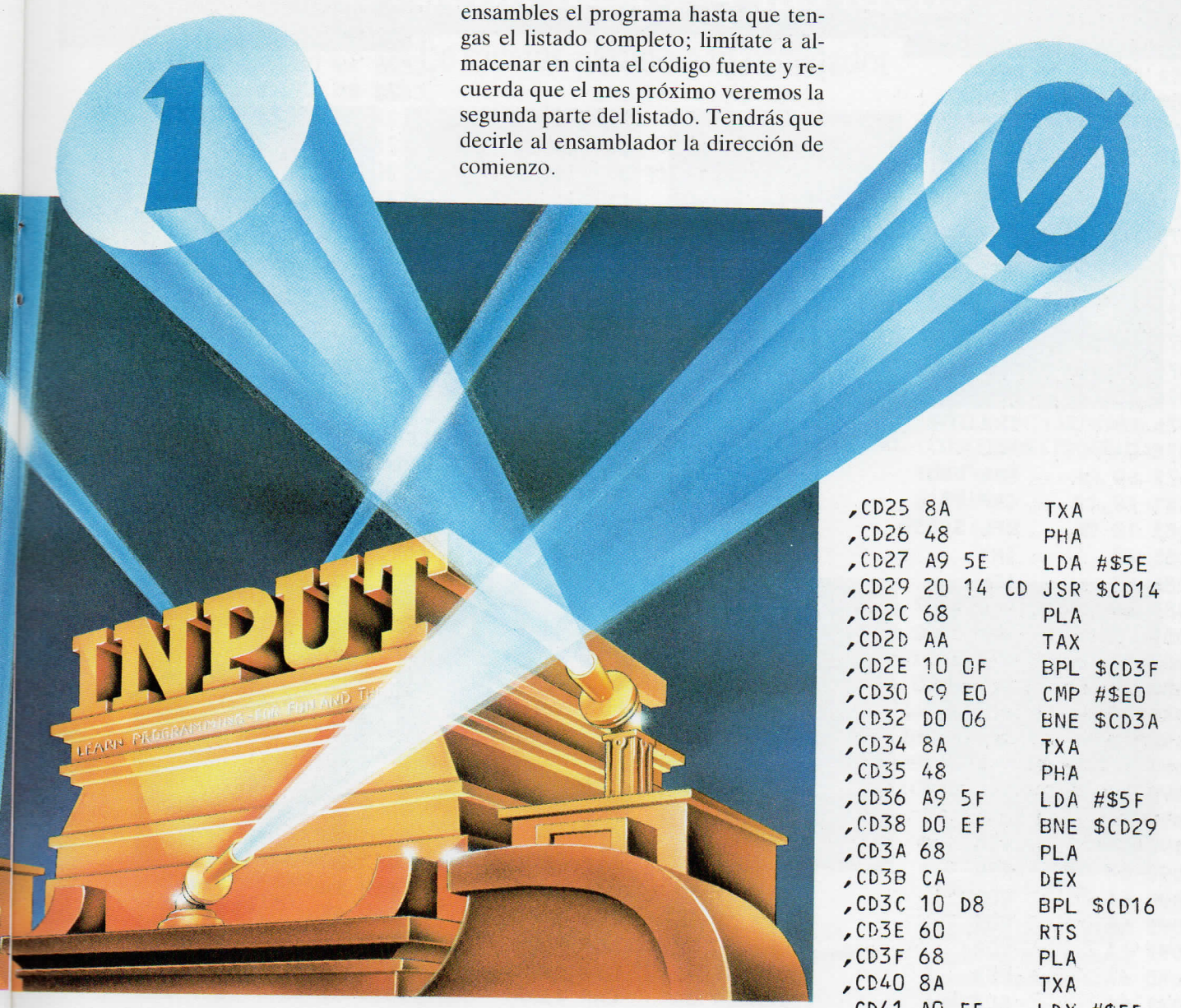
Si no dispones de un ensamblador comercial, tienes que introducir en tu máquina un corto programa en BASIC, ejecutarlo e introducir todo el código hexadecimal; el programa POKEará los valores hexadecimales en las direcciones de memoria adecuadas. Pero ya nos ocuparemos más detalladamente de esto la próxima vez.

El listado en lenguaje ensamblador

que figura a continuación es para el **Commodore 64**, pero si eres poseedor de un **Vic 20** con expansión, puedes adaptarlo con facilidad para que corra en tu máquina. En el listado de ensamblador las partes que han de ser modificadas se han escrito en minúscula, por lo que debes consultar la tabla para hallar la correspondiente equivalencia.

C-64	Vic 20			
	8K	16K	24K	32K
CD	3D	5D	7D	8D
CE	3E	5E	7E	8E
CF	3F	5F	7F	8F

ensambles el programa hasta que tengas el listado completo; limítate a almacenar en cinta el código fuente y recuerda que el mes próximo veremos la segunda parte del listado. Tendrás que decirle al ensamblador la dirección de comienzo.



Los que tengáis un **Vic 20** con 8, 16 o 24K de expansión, también tendréis que modificar dos punteros. En primer lugar hay que POKEar las dos direcciones de memoria.

```
Vic 20+8K
  POKE 52,61      POKE 56,61
Vic 20+16K
  POKE 52,93      POKE 56,93
Vic 20+24K
  POKE 52,125     POKE 52,125
```

El programa del **Commodore 64** empieza en CD14, por lo que si tienes un **Vic 20**, tendrás que cambiar el CD por el equivalente en tu máquina. No

```
C-64              CD14
Vic 20+8K         3D14
Vic 20+16K        5D14
Vic 20+24K        7D14
Vic 20+32K        8D14
```

Teclea para C-64 y Vic-20

```
,CD14 A2 1F     LDX #$1F
,CD16 48        PHA
,CD17 38        SEC
,CD18 FD 7D CE  SBC $CE7D,X
,CD1B F0 22     BEQ $CD3F
,CD1D C9 80     CMP #$80
,CD1F F0 04     BEQ $CD25
,CD21 C9 20     CMP #$20
,CD23 D0 0B     BNE $CD30
,CD25 8A        TXA
,CD26 48        PHA
,CD27 A9 5E     LDA #$5E
,CD29 20 14 CD  JSR $CD14
,CD2C 68        PLA
,CD2D AA        TAX
,CD2E 10 0F     BPL $CD3F
,CD30 C9 E0     CMP #$E0
,CD32 D0 06     BNE $CD3A
,CD34 8A        TXA
,CD35 48        PHA
,CD36 A9 5F     LDA #$5F
,CD38 D0 EF     BNE $CD29
,CD3A 68        PLA
,CD3B CA        DEX
,CD3C 10 D8     BPL $CD16
,CD3E 60        RTS
,CD3F 68        PLA
,CD40 8A        TXA
,CD41 A0 FF     LDY #$FF
,CD43 C9 0A     CMP #$0A
,CD45 F0 03     BEQ $CD4A
,CD47 8D 42 CD  STA $CD42
,CD4A AE 99 CD  LDX $CD99
,CD4D CA        DEX
,CD4E CA        DEX
,CD4F D9 9D CE  CMP $CE9D,Y
,CD52 D0 04     BNE $CD58
,CD54 A9 00     LDA #$00
,CD56 F0 31     BEQ $CD89
,CD58 CA        DEX
,CD59 D9 BE CE  CMP $CEBE,Y
,CD5C D0 04     BNE $CD62
,CD5E A9 40     LDA #$40
,CD60 D0 27     BNE $CD89
,CD62 C9 01     CMP #$01
,CD64 10 04     BPL $CD6A
```

```
,CD66 A9 60     LDA #$60
,CD68 10 1F     BPL $CD89
,CD6A CA        DEX
,CD6B CA        DEX
,CD6C CA        DEX
,CD6D CA        DEX
,CD6E CA        DEX
,CD6F 69 DF     ADC #$DF
,CD71 C9 F8     CMP #$F8
,CD73 10 14     BPL $CD89
,CD75 E8        INX
,CD76 0A        ASL
,CD77 69 07     ADC #$07
,CD79 C9 F0     CMP #$F0
,CD7B 10 0C     BPL $CD89
,CD7D E8        INX
,CD7E 0A        ASL
,CD7F 69 0F     ADC #$0F
,CD81 C9 C8     CMP #$C8
,CD83 10 04     BPL $CD89
,CD85 E8        INX
,CD86 0A        ASL
,CD87 69 37     ADC #$37
,CD89 C0 0B     CPY #$0B
,CD8B D0 05     BNE $CD92
,CD8D A0 20     LDY #$20
,CD8F 8C 42 CD  STY $CD42
,CD92 8E B5 CD  STX $CDB5
,CD95 AE 99 CD  LDX $CD99
,CD98 D0 00     BNE $CD9A
,CD9A 6A        ROR
,CD9B 6A        ROR
,CD9C 6A        ROR
,CD9D 6A        ROR
,CD9E 6A        ROR
,CD9F 6A        ROR
,CDA0 6A        ROR
,CDA1 6A        ROR
,CDA2 48        PHA
,CDA3 2A        ROL
,CDA4 3D E0 CE  AND $CEE0,X
,CDA7 A0 00     LDY #$00
,CDA9 11 61     ORA ($61),Y
,CDAB 91 61     STA ($61),Y
,CDAD C8        INY
,CDAE 68        PLA
,CDAF 3D E8 CE  AND $CEE8,X
,CDB2 91 61     STA ($61),Y
,CDB4 A9 FF     LDA #$FF
,CDB6 10 08     BPL $CDC0
,CDB8 49 F8     EOR #$F8
,CDBA E6 61     INC $61
,CDBC D0 02     BNE $CDC0
,CDBE E6 62     INC $62
,CDC0 8D 99 CD  STA $CD99
,CDC3 A9 00     LDA #$00
,CDC5 60        RTS
,CDC6 48        PHA
,CDC7 98        TYA
,CDC8 48        PHA
,CDC9 A0 08     LDY #$08
,CDCB 8C 1A CE  STY $CE1A
,CDCE A0 00     LDY #$00
,CDD0 8C 13 CE  STY $CE13
,CDD3 B1 2D     LDA ($2D),Y
,CDD5 C8        INY
,CDD6 C9 5A     CMP #$5A
,CDD8 D0 06     BNE $CDE0
,CDDA B1 2D     LDA ($2D),Y
,CDDC C9 80     CMP #$80
,CDDE F0 06     BEQ $CDE6
,CDE0 98        TYA
,CDE1 69 06     ADC #$06
,CDE3 A8        TAY
,CDE4 D0 ED     BNE $CDD3
,CDE6 C8        INY
,CDE7 8C 2F CE  STY $CE2F
,CDEA 68        PLA
,CDEB A8        TAY
,CDEC 68        PLA
,CDED 60        RTS
,CDEE 48        PHA
,CDEF 8A        TXA
,CDF0 48        PHA
,CDF1 98        TYA
,CDF2 48        PHA
,CDF3 A5 2F     LDA $2F
,CDF5 85 61     STA $61
,CDF7 A5 30     LDA $30
,CDF9 85 62     STA $62
,CDFB A0 00     LDY #$00
,CDFD B1 61     LDA ($61),Y
,CDFF C9 DA     CMP #$DA
,CE01 D0 07     BNE $CE0A
,CE03 C8        INY
,CE04 B1 61     LDA ($61),Y
,CE06 C9 80     CMP #$80
,CE08 F0 08     BEQ $CE12
,CE0A A0 02     LDY #$02
,CE0C 18        CLC
,CE0D 20 6B CE  JSR $CE6B
,CE10 90 E9     BCC $CDFB
,CE12 A9 FF     LDA #$FF
,CE14 A0 08     LDY #$08
,CE16 91 61     STA ($61),Y
,CE18 88        DEY
,CE19 A9 FF     LDA #$FF
,CE1B 91 61     STA ($61),Y
,CE1D 38        SEC
,CE1E 20 6B CE  JSR $CE6B
,CE21 A9 07     LDA #$07
,CE23 8D 99 CD  STA $CD99
,CE26 A9 00     LDA #$00
,CE28 8D 42 CD  STA $CD42
,CE2B 8D 40 CE  STA $CE40
,CE2E A0 FF     LDY #$FF
,CE30 B1 2D     LDA ($2D),Y
,CE32 8D 42 CE  STA $CE42
,CE35 C8        INY
,CE36 B1 2D     LDA ($2D),Y
,CE38 85 65     STA $65
,CE3A C8        INY
,CE3B B1 2D     LDA ($2D),Y
,CE3D 85 66     STA $66
,CE3F A0 FF     LDY #$FF
,CE41 C0 FF     CPY #$FF
,CE43 F0 0A     BEQ $CE4F
,CE45 B1 65     LDA ($65),Y
,CE47 20 14 CD  JSR $CD14
,CE4A EE 40 CE  INC $CE40
,CE4D D0 F0     BNE $CE3F
,CE4F 20 14 CD  JSR $CD14
,CE52 A9 06     LDA #$06
,CE54 CD 99 CD  CMP $CD99
,CE57 A5 61     LDA $61
,CE59 E5 63     SBC $63
,CE5B 8D 1A CE  STA $CE1A
,CE5E A5 62     LDA $62
,CE60 E5 64     SBC $64
,CE62 8D 13 CE  STA $CE13
,CE65 68        PLA
,CE66 A8        TAY
,CE67 68        PLA
,CE68 AA        TAX
,CE69 68        PLA
,CE6A 60        RTS
,CE6B A5 61     LDA $61
,CE6D 85 63     STA $63
,CE6F 71 61     ADC ($61),Y
,CE71 85 61     STA $61
,CE73 C8        INY
,CE74 A5 62     LDA $62
,CE76 85 64     STA $64
,CE78 71 63     ADC ($63),Y
,CE7A 85 62     STA $62
,CE7C 60        RTS
```

En la linea 70 de la página 68 del coleccionable aparece una pequeña errata. Debe teclearse como sigue:

```
70 PRINT "[CLR/HOME][CTRL+6]
   ;SPC(30);"PATO:";TU
```

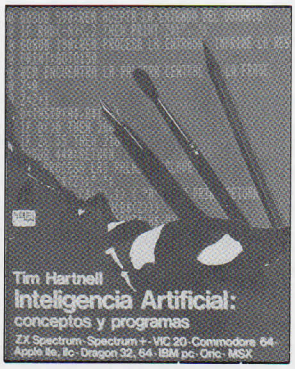
Ordena tus propias ideas

Le sacarás partido a tu ordenador

EL LIBRO GIGANTE DE LOS JUEGOS PARA COMMODORE 64
Gregg Barnett
1.484 ptas.

EL SUPERLIBRO DE LOS JUEGOS PARA ORDENADOR
Tim Hartnell
2.120 ptas.

SISTEMAS EXPERTOS. INTRODUCCION AL DISEÑO Y APLICACIONES
Tim Hartnell
2.120 ptas.

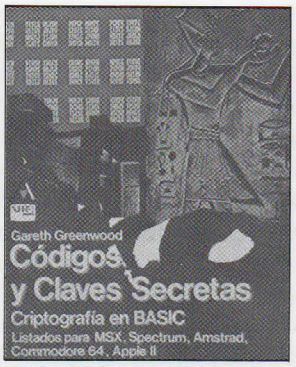


INTELIGENCIA ARTIFICIAL CONCEPTOS Y PROGRAMAS
Tim Hartnell
1.484 ptas.

SIMULACIONES: REPLICA LA REALIDAD CON TU ORDENADOR
Tim Hartnell
1.643 ptas.

PASCAL A PARTIR DEL BASIC
Peter Brown
1.590 ptas.

PROGRAMACION AVANZADA DEL COMMODORE 64.
Ampliación del Basic y rutinas gráficas
John P. Gibbons
1.802 ptas.



CODIGOS Y CLAVES SECRETAS: CRIPTOGRAFIA EN BASIC
Listados para MSX, Spectrum, Amstrad, Commodore 64, Apple II
Gareth Greenwood
1.378 ptas.

☐ Les ruego me envíen el catálogo de su editorial.
☐ Les ruego me envíen los siguientes títulos:

_____

_____

TOTAL _____

☐ Adjunto talón bancario a
GRUPO DISTRIBUIDOR EDITORIAL, S. A.

☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).

☐ Giro postal.

Nombre _____

Profesión _____

Dirección _____

C. P. _____ Localidad _____

Provincia _____

IC

ANAYA MULTIMEDIA

Adquiéralos en su librería habitual.
Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a:
Apdo. de Correos 14632, Ref. D. de C. 28080 MADRID

MODELIZANDO LA REALIDAD

- TIPOS DE MODELOS
- INCERTIDUMBRES
- DISTRIBUCION UNIFORME
- DISTRIBUCION EXPONENCIAL
- DISTRIBUCION NORMAL

Elegir la combinación ganadora en un juego de azar es equivalente a la extracción de números aleatorios. Es una tarea adecuada para tu micro, siempre que poseas unas nociones de estadística.

Los ingenieros de todas las especialidades construyen modelos, estructuras en miniatura, para probar sus diseños. Al hacer esto están simulando un sistema real. En el artículo de simulación, que presentamos hace unos meses, vimos que tu micro puede hacer algo semejante, como es ir recorriendo las etapas sucesivas de un determinado proceso para dar un resultado que se parezca a algo que ocurre realmente. Aquí te presentamos la manera de combinar los principios de la estadística y de la modelización para resolver problemas de una forma realista, tanto en juegos como en situaciones de la vida real.

TIPOS DE MODELOS

Técnicamente un modelo es una representación de un sistema real, que se construye por su sencillez y utilidad como ayuda en la toma de decisiones. Hay tres categorías de modelos: iconográficos, analógicos y simbólicos, aunque no todos resultan practicables en la simulación por parte de los entusiastas del micro doméstico.

Los modelos iconográficos no tienen partes móviles. A esta categoría pertenecen los mapas de carreteras y los aviones de juguete. Por ejemplo, en los estudios previos para la construcción de un puente sobre un estuario, los ingenieros de caminos podrían construir un modelo del puente y de la zona que rodea al estuario, para estudiar los efectos de la marea, las riadas y el viento sobre la estructura. Aunque se emplean ordenadores para monitorizar estos factores y analizar los resultados, el hecho de tener que recopilar datos a partir de ensayos reales significa que no hay necesidad de realizar una simulación.

En los modelos analógicos se representa una cantidad por otra. Por ejemplo el profesor **Phillips**, un economista británico, dispuso unos cuantos depósitos llenos de agua fría a diferentes alturas y utilizó el flujo de líquido entre los tanques para representar el flujo monetario en la economía del Reino Unido. Puede hacerse lo mismo electrónicamente, representando las diferentes cantidades por medio de diferentes valores de tensión eléctrica, pero para esto se emplean generalmente ordenadores analógicos, en lugar de los digitales, como es un micro doméstico. La tercera categoría, el modelo simbólico o abstracto, es la que resulta especialmente interesante para el que quiera llevar a cabo simulaciones en un ordenador doméstico.

INCERTIDUMBRES DE LA VIDA REAL

En los modelos de tipo simbólico, se utiliza una fórmula matemática para representar el sistema real. Por ejemplo, si quieres estudiar la distancia (S) recorrida por un coche que viaja a velocidad constante (V) durante un cierto tiempo (T), el modelo adecuado sería S = V*T. Este tipo de

ecuación se llama determinística, debido a que no da ninguna medida de las incertidumbres. Describe con precisión el movimiento del automóvil para cualquier conjunto de valores que especifiques. Sin embargo existen muchos sucesos que no pueden ser especificados sin un cierto grado de incertidumbre. Tienes un ejemplo de esto en el lanzamiento al aire de una moneda. Los modelos que describen este tipo de sucesos imprecisos se llaman estocásticos. Normalmente son este tipo de sucesos los que requieren simulación, particularmente en los juegos.

Algunas variables pueden tomar solamente valores discretos, tales como 0, 1, 2, 3, ... A esta categoría pertenecen el número de goles en un partido de fútbol o el número de hijos varones de una familia. Cuando se simulan variables aleatorias de este tipo, es importante darse cuenta de que existen muchos procesos aparentemente diferentes que tienen de hecho la misma estructura subyacente.

Consideremos por ejemplo el caso de la simulación del lanzamiento de una moneda al aire, anteriormente mencionado. Este programa podría modificarse fácilmente para representar los resultados aleatorios de una partida de dados o el número de goles conseguidos en una serie de lanzamientos de *penalties*. En todos estos casos hay un cierto número de pruebas repetidas independientes, cuyos resultados pueden producirse con probabilidades bien definidas. Supongamos que la prueba consiste en lanzar una moneda al aire cuatro veces. El resultado de cada prueba puede ser cuatro caras, tres caras y una cruz, dos caras y dos cruces, una cara y tres cruces o cuatro cruces. Además hay una probabilidad bien definida asociada con cada uno de estos posibles resultados. Las estructuras de este tipo, llamadas procesos de **Bernoulli**, generan variables aleatorias discretas que no resultan adecuadas, por ejemplo, para el estudio del número de accidentes que ocurren en un **Grand Prix**, el número total de llamadas telefónicas que se producen en un período de media hora o el número de rayos registrado durante una tormenta eléctrica. En todos estos ejemplos, los sucesos se producen de modo casual durante un determinado período de tiempo, ellos solos y no como consecuencia de un conjunto de pruebas repetidas. A estas estructuras se les llama procesos de **Poisson**. Teclea el siguiente programa que te permite realizar un muestreo a partir de cualquier distribución de **Poisson** que elijas:

Teclea para C-64 y Vic-20

```
40 PRINT"[SHIFT+CLR/HOME]
   [CRSR abajo]>[CRSR izq.]
   [CTRL+9]SIMULACION DE
   POISSON[2*CRSR abajo]"
50 PRINT"[CRSR izq.]CUAL ES
   TU VALOR MEDIO ":INPUT A
70 INPUT "[CRSR abajo]TAMA#O
   DE LA MUESTRA ";N
80 FOR I=1 TO N
90 C=0:T=1
100 S=EXP(-A)
110 T=T*RND(1)
120 IF T<=S THEN PRINT C,:
    GOTO 150
130 C=C+1
140 GOTO 110
150 NEXT I:PRINT
160 PRINT"OTRA MUESTRA (S/N)
    ?"
170 GET G$:IF G$="S" THEN
    RUN
180 IF G$<>"N" THEN 170
190 PRINT"[SHIFT+CLR/
    HOME]"
```

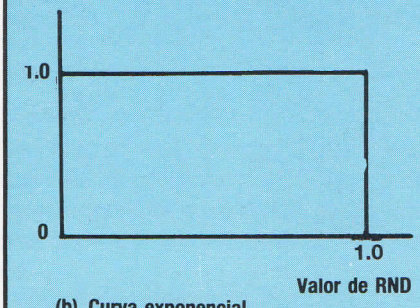
Al ejecutar el programa, se te pide que introduzcas un valor promedio (la media aritmética) y el tamaño de tu muestra (líneas 50 y 70). La parte principal del programa (líneas 80 a 150) utiliza una ecuación para generar las variables de **Poisson**. A la variable S se le asigna (línea 100) el valor del número e (una constante matemática) elevado a la potencia A (el valor de la media que tú has introducido). La línea 110 elige un número aleatorio entre 0 y el valor de T, y realiza un cambio de escala según el valor de T. La línea 120 compara S y T para decidir si hay que presentar o no una variable (C).

Supongamos por ejemplo que estás diseñando un juego espacial en el que una nave interplanetaria tiene que atravesar una tormenta de meteoritos. Si el número medio de «impactos» en cualquier período de un minuto es dos y se requieren cinco simulaciones separadas de un minuto cada una, el ordenador podría presentar los números 2, 3, 2, 1, 0 como estadística de impactos. Cada elemento de la lista no será muy diferente de 2 (la media), y en total hay cinco elementos (el tamaño de la muestra).

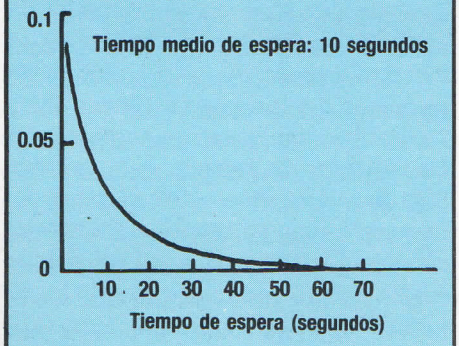
¿Podría utilizarse esta misma técnica para simular, digamos, el tanteo final de un partido de fútbol? Está claro que se trata de una variable discreta, ya que el número de goles conseguidos sólo puede tomar valores enteros: 0, 1, 2, 3, etc. Además son sucesos que se producen por sí solos en el tiempo y no como resultado de pruebas repetidas. En consecuencia se trata de un caso susceptible de ser adecuadamente descrito mediante un pro-

Comparación de las distribuciones

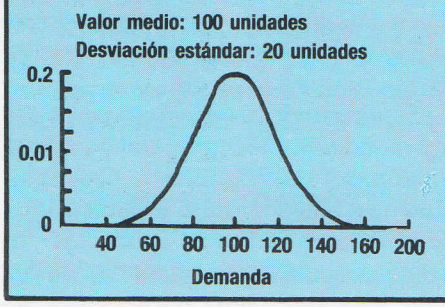
(a) Distribución uniforme



(b) Curva exponencial



(c) Curva normal



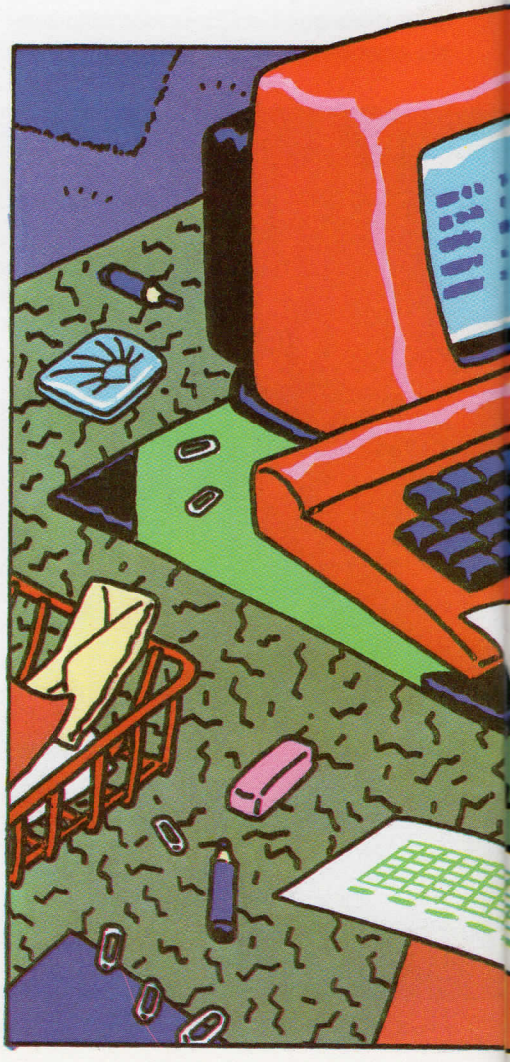
ceso de **Poisson**. Aparte del número de partidos jugados, la única información complementaria que necesitas en este caso es qué valor hay que utilizar como número medio de goles. En este punto se hace necesario salir al mundo real y recoger algunos datos.

Tomando como base los goles marcados en y fuera de casa durante la Liga una elección razonable para los tanteos sería adoptar el valor 1.7 como media de goles domésticos y 1.0 como media de goles fuera de casa. La división por dos de cada uno de estos valores conduce a un tanteo medio por cada medio partido de 0.85 y 0.5. La construcción del programa resulta ya inmediata:

Teclea para C-64 y Vic-20

```
20 DIM A$(25),H$(25),H(50),
   A(50),FA(25),FH(25)
50 PRINT"[SHIFT+CLR/HOME]
   [CRSR abajo]>[CRSR izq.]
   [CTRL+9]PUNTUACION FINAL
   [2*CRSR abajo]"
70 PRINT"CUANTOS PARTIDOS
   (1-25)":INPUT N
75 IF N<1 OR N>25 THEN GOTO
   70
80 FOR I=1 TO N
90 PRINT"PARTIDO ",I
100 INPUT "EQUIPO DE CASA ";
    H$(I)
110 INPUT "EQUIPO VISITANTE"
    ;A$(I):PRINT
120 NEXT I
130 FOR V=1 TO 1000:NEXT V
150 FOR I=1 TO 2*N
160 C=0:T=1
170 S=EXP(-.85)
180 T=T*RND(1)
190 IF T<=S THEN H(I)=C:GOTO
    220
200 C=C+1
210 GOTO 180
220 NEXT I
230 FOR I=1 TO 2*N
240 C=0:T=1
250 S=EXP(-.5)
260 T=T*RND(1)
270 IF T<=S THEN A(I)=C:GOTO
    300
280 C=C+1
290 GOTO 260
300 NEXT I
320 PRINT"[SHIFT+CLR/HOME]
    [CRSR abajo]>[CRSR izq.]
    [CTRL+9]GOLES DURANTE EL
    PRIMER TIEMPO[2*CRSR
    abajo]"
330 FOR I=1 TO N
340 PRINT H$(I);H(I):PRINT
    "[CTRL+9]";A$(I);A(I):
    PRINT
350 FOR D=1 TO 200:NEXT D,I
360 PRINT"PRESIONA UNA TECLA
    PARA CONTINUAR"
370 GET X$:IF X$=""THEN 370
380 PRINT"[SHIFT+CLR/HOME]
    [CRSR abajo]>[CRSR izq.]
    [CTRL+9]GOLES FINALES
    [2*CRSR abajo]"
390 FOR I=1 TO N
400 FH(I)=H(I)+H(N+1)
410 FA(I)=A(I)+A(I+N)
420 PRINT H$(I);FH(I):PRINT
    "[CTRL+9]";A$(I);FA(I):
    PRINT
430 FOR D=1 TO 200:NEXT D,I
440 PRINT"[CRSR abajo]OTRA
    VUELTA (S/N)?"
450 GET G$:IF G$="N" THEN
    PRINT"[SHIFT+CLR/HOME]":
    END
455 IF G$<>"S" THEN 450
460 INPUT "[SHIFT+CLR/HOME]
    LOS MISMOS EQUIPOS
    (S/N)";P$
470 IF P$="N" THEN RUN
480 GOTO 150
```

Ejecuta el programa y responde a las preguntas que te hace su primera sección (desde el principio hasta la lí-



nea 120). Para evitar el tener que teclear demasiado, puedes introducir letras del alfabeto en vez de nombres reales para ambos equipos. Las líneas 150 a 300 utilizan el algoritmo de **Poisson** que vimos en el primer programa, para generar los tanteos del medio tiempo en casa y fuera de casa. Las líneas 320 a 350 organizan la presentación de las puntuaciones de medio tiempo y las líneas 390 a 430 se ocupan del resultado de los partidos completos.

Aunque los goles individuales simulados por partido no correspondan con los resultados reales, el comportamiento global de la situación simulada debería parecerse bastante a dichos resultados reales.

La distribución de **Poisson** funciona bien con variables que tomen valores enteros; pero hay otras variables aleatorias continuas que toman valores fraccionarios. Por ejemplo en una simulación de ordenador del salto de longitud en una **Olimpiada**, un participante podría saltar una distancia cualquiera entre seis y nueve metros. La longitud de dicho salto es un ejemplo de variable aleatoria continua.

Aunque la función RND del ordenador es adecuada para la simulación de variables aleatorias discretas de tipo **Poisson**, necesitas una función RND modificada para la modelización con variables continuas. La figura 1 compara el tipo de distribución dado por ambos tipos de variables. Las variables discretas dan una distribución uniforme, mientras que las variables continuas pueden dar una distribución exponencial o normal. Teclea el siguiente programa para transformar las variables aleatorias planas o uniformemente distribuídas que te da la función RND de tu ordenador en variables aleatorias con distribución exponencial.

Teclea para C-64 y Vic-20

```
40 PRINT"[SHIFT+CLR/HOME]
   [CRSR abajo]>[CRSR izq.]
   [CTRL+9]SIMULACION":PRINT
   "2*CRSR izq.][CTRL+9]
   EXPONENCIAL[2*CRSR abajo]
50 PRINT "CUAL ES TU VALOR":
   INPUT"MEDIO ";A
70 INPUT "[2*CRSR abajo]
   TAMA#O DE LA MUESTRA";N
80 FOR I=1 TO N
90 X=(-A)*LOG(RND(1)):Y=INT
   (X*10):PRINT .1*Y,
100 NEXT I
110 PRINT:PRINT"OTRA MUESTRA
   (S/N)?"
120 GET G$:IF G$="S" THEN
   RUN
130 IF G$<>"N" THEN 120
140 PRINT"[SHIFT+CLR/HOME]"
```

Al ejecutar este programa tienes que especificar el valor medio y el número de muestras requeridas; a continuación aparecerán en pantalla las variables. Todo se hace en la línea 90. Se utiliza una técnica matemática llamada método de la transformada inversa, para convertir variables «planas» en variables exponenciales. Este programa puede usarse también perfectamente para simular, por ejemplo, el tiempo de espera de una secadora en una lavandería, o la vida media de los paquetes de energía en un juego de aventura espacial. En todos los casos el concepto importante es el del tiempo transcurrido.

DISTRIBUCION NORMAL

La curva normal en forma de campana representa la más famosa de las distribuciones estadísticas. Ha resultado muy útil en la descripción de todo tipo de fenómenos naturales, incluyendo las tallas y los pesos de las personas adultas. Además está ampliamente aceptada como el medio más apropiado de describir la distribución de errores.

Para ilustrar la forma en que podría incorporarse la distribución normal

dentro de un modelo, consideremos el siguiente ejemplo para diseñar un juego comercial. Supongamos que las ventas (S) de una empresa dependen de algún nivel base (S0) y de la cantidad que se ha gastado en publicidad (A). Esta relación podría resumirse con la expresión S = S0 + (B*A), siendo B una constante conocida.

Pero en la vida real raramente se dan estas relaciones tan precisas. Los sucesos y circunstancias imprevistos distorsionan los hechos o introducen «ruido» en el sistema. Este sistema se puede modelizar adecuadamente introduciendo una ecuación de perturbación con distribución normal para las ventas. La ecuación completa para la cifra de ventas sería entonces S = S0 + (B*A) + Distribución normal.

Para ver cómo funciona, teclea el siguiente programa:

Teclea para C-64 y Vic-20

```
40 DIM U(50),Y(50),Z(50)
50 PRINT "[SHIFT+CLR/HOME]
   [CRSR abajo]>[CRSR izq.]
   [CTRL+9]SIMULACION NORMAL
   [2*CRSR abajo]"
60 PRINT"CUAL ES TU VALOR":
   INPUT"MEDIO ";A
70 PRINT"[CRSR abajo]CUAL ES
   TU VALOR":INPUT "MINIMO "
   ;M
80 INPUT "TAMA#O DE LA
   MUESTRA ";N
90 B=(A-M)/2.5
100 FOR I=1 TO N
110 T=0
120 FOR J=1 TO 15
130 Y(J)=RND(1)
140 T=T+Y(J):NEXT J
150 U(I)=((T/15)-.5)/SQR(1/
   (12*15))
160 Z(I)=A+(B*U(I))
170 P=INT(10*Z(I)):PRINT .1
   *P,
180 NEXT I
190 PRINT:PRINT"OTRA MUESTRA
   (S/N)?"
200 GET G$:IF G$="S" THEN
   RUN
210 IF G$<>"N" THEN 200
220 PRINT"[SHIFT+CLR/HOME]"
```

Ejecuta la simulación y especifica los valores medio y mínimo de tu variable. Técnicamente, el margen de valores de la curva normal no está acotado, por lo que existe una pequeña probabilidad de que un valor simulado sea menor que el valor mínimo especificado por tí. Pero esto no sucederá muy a menudo.

Las líneas 120 a 150 utilizan la función RND para generar 15 números aleatorios entre 0 y 1 y a continuación sumarlos. La línea 160 calcula la media y le añade un factor de escala. Después de otro cambio de escala (línea 160) se imprime el resultado.

Tal vez te ayude a tener una sensación más real de lo que ocurre, el hecho de hacer que los números se generen a partir de dos distribuciones con la misma media y aproximadamente el mismo margen de valores, pero con diferentes formas. Ejecuta este programa introduciendo los valores de 100 para la media, 50 para el valor mínimo y 40 para el tamaño de la muestra. Con los valores obtenidos puedes hacer un gráfico parecido al de la figura 1C.

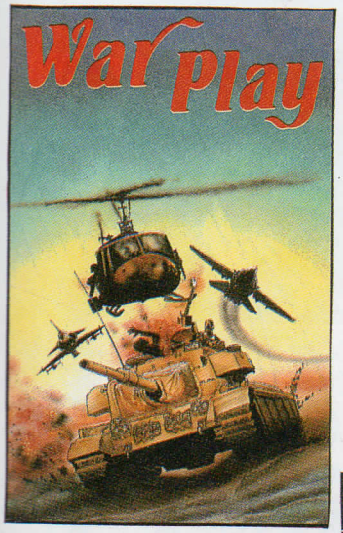
Borra ahora la línea 90 y las líneas 120 a 160, y escribe de nuevo la siguiente línea:

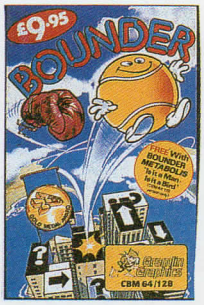
```
110 Z(I)=M+RND(1)*2*(A-M)
```

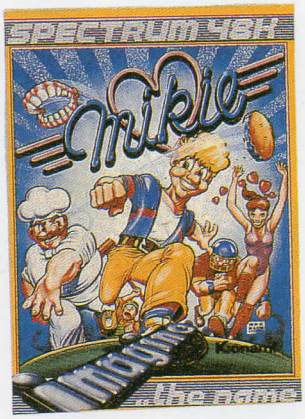
Estos cambios hacen que el programa sea capaz de presentar valores aleatorios con distribución uniforme. Ejecuta de nuevo el programa, introduciendo los mismos valores que antes y compara los resultados. Al dibujar esta vez los valores se obtiene un gráfico como el de la figura 1B. Observa que las lecturas normales (las de la primera prueba) están mucho más apiñadas en torno al valor medio.

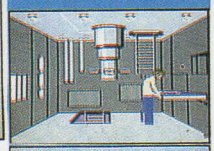
COMPULAND SIEMPRE POR DELANTE

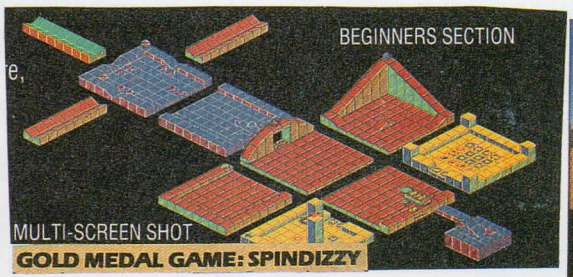
CON VISTAS A LA INMINENTE LLEGADA DE LAS VACACIONES ESTE MES
OFRECEMOS UNA PEQUENA MUESTRA DE LAS ULTIMAS NOVEDADES

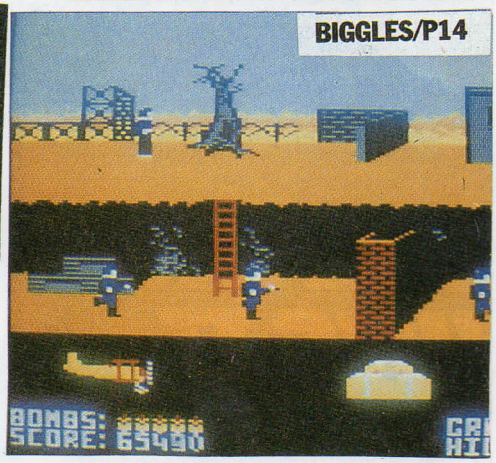


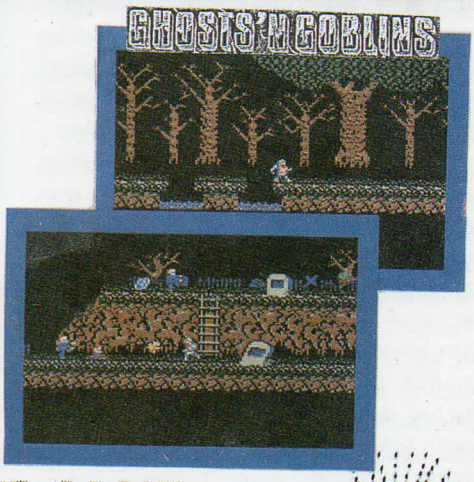


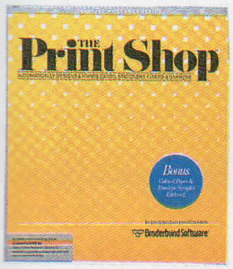












CALVO ASENSIO N.8 MADRID 28015

TFN.2431638 CATALOGO CON MAS DE 1000 PRODUCTOS !!PEDIRLO!!
PERTENECIENDO AL CLUB DE SOFTWARE TENDRAS UN 10% DE DESCUENT

PON MUSICA MIENTRAS TRABAJAS

- EJECUCION Y PROCESAMIENTO SIMULTANEOS
- PROGRAMACION EN CODIGO MAQUINA
- USO DE LAS INTERRUPCIONES
- LA CIRCUITERIA DEL SONIDO

Si te gusta escuchar una suave melodía de fondo mientras programas, te agradará este sencillo programa en código máquina que se activa mediante interrupciones, y que además no agotará la memoria disponible.

La mayor parte de los buenos juegos que puedas comprar, llevan incorporados impresionantes efectos sonoros o musicales para ofrecer mayor atractivo. Aquí tienes un programa que hará posible que tu ordenador produzca música mientras tecleas un programa o desarrollas uno propio.

Casi todos los programas de juego tienen una introducción musical o un breve pasaje para destacar determinados sucesos, por ejemplo cuando haces un buen tanteo o pierdes una vida. Sin embargo, habrás observado que el efecto más impresionante es que las melodías no se interpretan momentáneamente, sino a lo largo de todo el juego. La razón por la que el **64** puede ejecutar un programa e interpretar música al mismo tiempo es que lleva incorporado un *chip* generador de sonido que trabaja independientemente del procesador central, a diferencia de otros micros, tal como el **Spectrum**, que se sirven del procesador central para producir el sonido.

Para el **Spectrum** hay disponibles unos cuantos juegos que permiten que suene una música constantemente a lo largo de todo el programa. Pero, para conseguir este efecto hay que proceder de forma muy cuidadosa jugando con los tiempos, de forma que parezca que están ocurriendo simultáneamente dos cosas. En general esto no suele resultar práctico en un programa de uso general.

El programa funciona por medio de interrupciones, por lo que a intervalos regulares se detecta un determinado suceso, haciendo que el programa salte a una rutina para ejecutar la melodía. En el caso de que quieras tener un momento de silencio, por ejemplo cuando suena el teléfono, tendrás que bajar el volumen del sonido en tu televisor.

Introduce el programa, pero date cuenta que contiene una sección de código máquina (contenida en las sentencias DATA), por lo que debes hacer una copia en cinta o disco antes de ejecutarlo.

Teclea para Commodore-64

```
1000 S=0:FORN=49152 TO 49407
     :READ A:POKE N,A:S=S+A:
     NEXT N
1050 IF S<>5666 THEN 1400
1100 S=0:FORN=24576 TO 24631
     :READ A:POKE N,A:S=S+A:
     NEXT N
1150 IF S<>6270 THEN 1500
1200 S=0:FORN=28672 TO 28778
     :READ A:POKE N,A:S=S+A:
     NEXT N
1250 IF S<>12659 THEN 1600
1300 PRINT "[SHIFT+CLR/HOME]"
     :SYS 24576: END
1400 PRINT "COMPRUEBA LAS
     LINEAS 2000-2060":END
1500 PRINT "COMPRUEBA LAS
     LINEAS 2100-3020":END
1600 PRINT "COMPRUEBA LAS
     LINEAS 4000-4050":END
2000 DATA 37,17,37,3,37,17,
     27,3,47,16,7,3,37,17,7,
     3,63,19,62,8
2010 DATA 37,17,7,3,63,19,17
     ,3,37,17,7,3
2020 DATA 63,19,7,3,37,17,17
     ,3,47,16,7,3,110,15,72,8
2030 DATA 63,19,37,3,63,19,
     27,3,42,18,7,3
2040 DATA 63,19,7,3,154,21,
     62,8
2050 DATA 63,19,7,3,154,21,
     17,3,63,19,7,3
2060 DATA154,21,7,3,63,19,17
     ,3,42,18,7,3,37,17,72,8
2100 DATA 37,17,37,3,37,17,
     27,3,47,16,7,3,37,17,7,
     3,63,19,62,8
2110 DATA 37,17,7,3,63,19,17
     ,3,37,17,7,3
2120 DATA 63,19,7,3,37,17,17
     ,3,114,11,7,3,63,19,72
     ,8
2130 DATA 216,12,7,3,107,14,
     7,3,70,15,7,3,37,17,7,3
2140 DATA 154,21,27,3,63,19,
     7,3,63,19,7,3,37,17,42
     ,8
2150 DATA 47,16,7,3,37,17,7,
     3,63,19,37,3,154,21,37,
     3,227,22,72,8
2160 DATA 0,0,1,1,0,0,1,1,0,
     0,1,1,0,0,1,1
2170 DATA 0,0,1,1,0,0,1,1,0,
     0,1,1,0,0,1,1
2180 DATA 0,0,1,1,0,0,1,1,0,
     0,1,1,0,0,1,1
2190 DATA 0,0,1,1,0,0,1,1,0,
     0,1,1
3000 DATA 169,0,162,25,157,0
     ,212,202,208,250,169,15
     ,141,24,212,169,1,141
3010 DATA 1,195,169,0,141,2,
     195,141,4,195,169,32,
     141,3,195,169,246,141
3020 DATA 5,195,169,64,141,5
     ,212,120,169,0,141,20,3
     ,169,112,141,21,3,88,96
4000 DATA 174,4,195,173,13,
     220,41,1,240,3,32,16,
     112,76,49,234,173,2,195
4010 DATA 240,17,206,0,195,
     208,11,169,0,141,2,195,
     173,3,195,141,4,212
4020 DATA 96,206,1,195,240,
     13,169,1,205,1,195,208,
     5,169,0,141,6,212,96
4030 DATA 189,0,192,141,0,
     212,232,189,0,192,141,1
     ,212,232,189,0,192,141
4040 DATA 0,195,232,189,0,192
     ,141,1,195,232,142,4,195
     ,173,5,195,141,6,212
```

```
4050 DATA 172,3,195,200,140,
     4,212,169,1,141,2,195,
     96,0
```

Al ejecutar el programa en BASIC, se suman los números de las sentencias DATA y en caso de que hayas cometido un error aparecerá un mensaje que te enviará a las líneas que debes comprobar.

Los datos se leen y comprueban en tres bloques, por lo que es importante que los introduzcas como están listados ya que en caso contrario la comprobación dará error. La melodía está planificada de forma que se repite en un bucle sin solapamientos ni saltos. Esto requiere un cuidadoso tratamiento de los tiempos de las notas en relación con el período de interrupción. Además los datos de la melodía están codificados por lo que no aparecen como los valores normales que introducirías para ejecutar una canción. Todo esto significa que tienes pocas posibilidades de modificar la melodía a menos que estés familiarizado con las técnicas de programación por interrupciones.

Una vez que el programa ha sido ejecutado, la llamada a código máquina para ejecutar la música se produce automáticamente. Puedes hacer entonces un NEW para desembarazarte del BASIC y la melodía no resultará afectada. Para detener la ejecución, pulsa RUN/STOP y RESTORE, y para volver a arrancarla escribe SYS 24576.

EXPLICACION DE LAS SUBRUTINAS

Los números contenidos entre las DATA de las líneas 2010 a 3020 y 4000 a 4050 corresponden a dos subrutinas que se encargan de hacer que funcione la generación de música por interrupciones.

La primera subrutina se carga entre las direcciones de memoria $24576 ($6000) y 24631 ($6037). La segunda entre 28672 ($7000) y 28778 ($7069).

En las direcciones $6000 a $6009 está contenido un bucle que inicializa los registros del *chip* SID, poniéndolos a cero. Desde aquí hasta la $602A se realizan las operaciones de puesta de los valores iniciales del *chip* de sonido, dejándolo listo para que comience a trabajar. La subrutina de las líneas $602B a $6037 se encarga de manipular las interrupciones del **Commodore 64**. Aunque es extremadamente corta, merece la pena verla con algún detenimiento por su gran utilidad en otras muchas aplicaciones.

Trabajando en BASIC es detenida la tarea que ejecuta el ordenador cincuenta veces por segundo. En cada

Si quieres evitarte teclear las DATA comprendidas entre las lineas 2010 a 3020, puedes emplear un ensamblador o monitor de codigo maquina para introducir lo siguiente:

Direccion decimal de inicio: 24576

```
,6000 A9 00    LDA #$00
,6002 A2 19    LDX #$19
,6004 9D 00 D4 STA $D400
,X
,6007 CA       DEX
,6008 D0 FA    BNE $6004
,600A A9 0F    LDA #$0F
,600C 8D 18 D4 STA $D418
,600F A9 01    LDA #$01
,6011 8D 01 C3 STA $C301
,6014 A9 00    LDA #$00
,6016 8D 02 C3 STA $C302
,6019 8D 04 C3 STA $C304
,601C A9 20    LDA #$20
,601E 8D 03 C3 STA $C303
,6021 A9 F6    LDA #$F6
,6023 8D 05 C3 STA $C305
,6026 A9 40    LDA #$40
,6028 8D 05 D4 STA $D405
,602B 78       SEI
,602C A9 00    LDA #$00
,602E 8D 14 03 STA $0314
,6031 A9 70    LDA #$70
,6033 8D 15 03 STA $0315
,6036 58       CLI
,6037 60       RTS
```

Utiliza un ensamblador o monitor de codigo maquina en sustitucion de las lineas 4000 a 4050 del listado en BASIC.

Direccion decimal de inicio: 28672

```
,7000 AE 04 C3 LDX $C304
,7003 AD 0D DC LDA $DC0D
,7006 29 01    AND #$01
,7008 F0 03    BEQ $700D
,700A 20 10 70 JSR $7010
,700D 4C 31 EA JMP $EA31
,7010 AD 02 C3 LDA $C302
,7013 F0 11    BEQ $7026
,7015 CE 00 C3 DEC $C300
,7018 D0 0B    BNE $7025
,701A A9 00    LDA #$00
,701C 8D 02 C3 STA $C302
,701F AD 03 C3 LDA $C303
,7022 8D 04 D4 STA $D404
,7025 60       RTS
,7026 CE 01 C3 DEC $C301
,7029 F0 0D    BEQ $7038
,702B A9 01    LDA #$01
,702D CD 01 C3 CMP $C301
,7030 D0 05    BNE $7037
,7032 A9 00    LDA #$00
,7034 8D 06 D4 STA $D406
,7037 60       RTS
,7038 BD 00 C0 LDA $C000,X
,703B 8D 00 D4 STA $D400
,703E E8       INX
,703F BD 00 C0 LDA $C000,X
,7042 8D 01 D4 STA $D401
,7045 E8       INX
,7046 BD 00 C0 LDA $C000,X
,7049 8D 00 C3 STA $C300
,704C E8       INX
,704D BD 00 C0 LDA $C000,X
,7050 8D 01 C3 STA $C301
,7053 E8       INX
,7054 8E 04 C3 STX $C304
,7057 AD 05 C3 LDA $C305
,705A 8D 06 D4 STA $D406
,705D AC 03 C3 LDY $C303
,7060 C8       INY
,7061 8C 04 D4 STY $D404
,7064 A9 01    LDA #$01
,7066 8D 02 C3 STA $C302
,7069 60       RTS
```

una de esta interrupciones —producidas por un temporizador interno del ordenador— se realizan una serie de funciones, tales como la exploración del teclado para comprobar si ha sido pulsada una tecla, por ejemplo. Cuando el aludido reloj reclama una de sus interrupciones, el microprocesador salta a la subrutina cuya dirección está contenida en las direcciones $0314 y $0315, que se encarga del manejo de las interrupciones (todas esas tareas complementarias antes mencionadas). Una vez concluido, el microprocesador retorna a su anterior trabajo. Este funcionamiento es totalmente transparente para el usuario, que ni siquiera percibe su existencia.

Pués bien, la instrucción SEI, de la dirección 602B crea una interrupción que impide que pueda producirse ninguna otra mientras no sé de la orden contraria con CLI (fuera interrupciones), como hacemos en la dirección $6030. La finalidad que se persigue con esto es rehuir una eventual interrupción causada por el sistema, que podría alterar la operación que veremos a continuación.

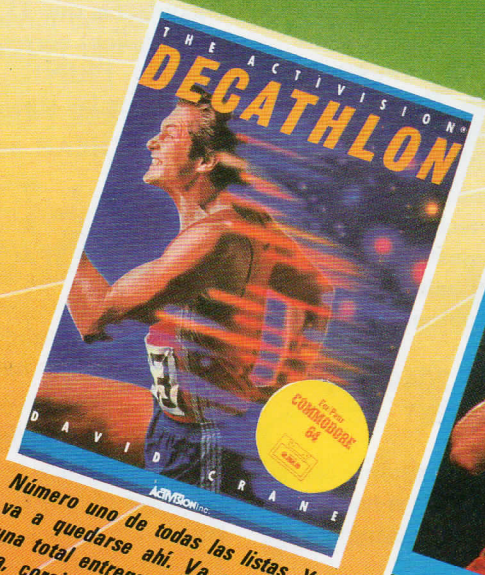
Estando seguros de que no existirán interferencias, cargamos una nueva dirección (vector) de comienzo para la rutina de interrupción, que en condiciones normales es la $EA31 (la dirección $0314 contiene $31 y la $0315 almacena $EA). Ahora conseguiremos que el microprocesador se dirija a la dirección $7000 cincuenta veces por segundo.

La subrutina que comienza en la dirección $7000 se encarga de leer los valores situados a partir de la dirección 49152 ($C000), que contienen la información que producirá la melodía que va volcándose en los registros del *chip* SID. El cambio de dichos valores se efectúa progresivamente, a medida que se producen las interrupciones. Después de cada uno de dichos ciclos, el programa salta a la dirección $EA31 (ver $700D), a partir de la cual la interrupción se trata del modo habitual. A este procedimiento se le denomina «introducir una cuña», de tal manera que en nuestro caso se produce la melodía libremente, sin interferir con el programa en BASIC.

Practica el mejor deporte con tu Commodore

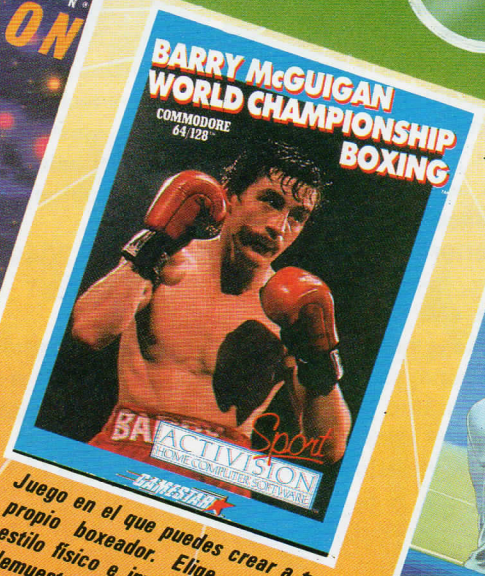
Entra en el mundo ciclista a través de las 16 etapas del Tour. Con acompañamiento musical y el jersey amarillo esperando al ganador.

C



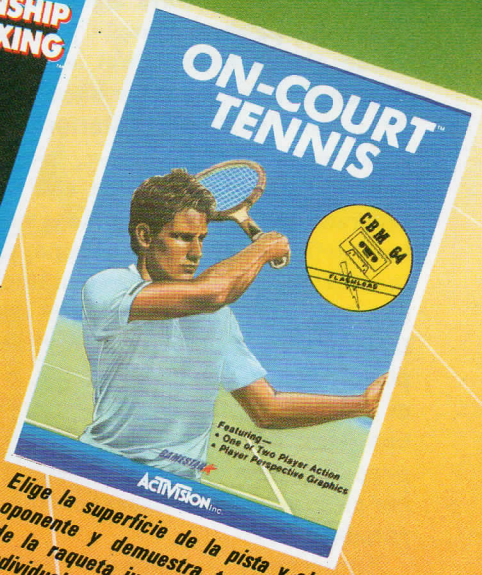
Número uno de todas las listas. Y va a quedarse ahí. Va a exigirle una total entrega. Velocidad, fuerza, coraje. El deseo de vencer.

CM



Juego en el que puedes crear a tu propio boxeador. Elige su raza, estilo físico e imagen. Entrénale y demuestra sus habilidades.

CS



Elige la superficie de la pista y el oponente y demuestra tu control de la raqueta jugando a dobles o individual.

C

Disponibles para:

COMMODORE	C
SPECTRUM	S
MSX	M

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES, O DIRECTAMENTE POR CORREO O TELEFONO A:

PROEIN, S.A.

Velázquez, 10 - 28001 Madrid Tels. (91) 276 22 08/09

COMO SE ALMACENAN LOS PROGRAMAS EN BASIC

La forma interna de trabajar el ordenador puede parecer a veces algo oscura, sobre todo cuando las cosas empiezan a no funcionar. Pero si consigues entender la manera en que se almacena un programa, estás en el buen camino para empezar a corregir sus errores.

¿Alguna vez te ha causado asombro lo que hace el ordenador con un programa cada vez que lo tecleas? Como probablemente sabes, el ordenador lo almacena en una zona especial de la memoria, reservada para los programas en BASIC, pero puede que no sepas dónde está situada dicha zona, ni qué aspecto presenta el programa cuando se convierte a una forma que resulte comprensible por el ordenador.

De hecho el programa almacenado en la memoria de tu ordenador se parece mucho a los caracteres que tecleas y muchos de los números que contiene la memoria son los códigos ASCII de las letras del programa. Pero el ordenador acorta algunas palabras —las **palabras clave**— y además añade alguna información adicional, situada literalmente entre líneas, a fin de poder localizar posteriormente dónde está cada cosa.

El ordenador dispone de una zona separada en la que almacena las variables definidas por el programa y, a medida que éstas cambian durante la ejecución, las va manteniendo constantemente actualizadas. Así, cuando por ejemplo si tecleas 10 LET A = 6, el ordenador almacena la línea como una parte de tu programa y después pone el nombre y el valor de la variable en el área de almacenamiento de variables.

Empecemos ya nuestra exploración. Primero tienes que introducir el siguiente programa. Tecléalo exactamente igual que te lo presentamos aquí. Cuida de no introducir espacios adicionales. Puedes ahora utilizar el programa para descubrir la manera en que el ordenador almacena sus programas.

Teclea para Commodore-64

```
10 PRINT "HOLA"
20 LET A=2
30 PRINT A
40 STOP
```

En el **Commodore 64** hay dos direcciones de memoria que conjuntamente actúan como puntero para señalar el comienzo del programa en BASIC. Estas direcciones son la 43 y la 44. Para saber dónde empieza tu programa en BASIC, teclea lo siguiente:

```
PRINT PEEK(43)+256*PEEK(44)
```

Normalmente obtendrás como respuesta el número 2049. Si te encuentras con que tu ordenador te da un valor diferente y has tecleado el comando correctamente, sustituye en el siguiente comando el valor 2049 por el número que hayas obtenido:

```
FOR I=2049-1 TO 2049+39:
PRINT PEEK(I);:NEXT
```

Deberás obtener la siguiente salida. De no ser así, comprueba si has tecleado el comando correctamente:

```
0  15  8  10  0  153  32  34
72  69  76  76  79  34  0
25  8  20  0  136  32  65
178  50  0  33  8  30  0  153
32  65  0  39  8  40  0  144
0  0  0
```

Utilizando ahora los códigos ASCII de las letras y los números, mira a ver si le encuentras algún sentido a lo siguiente:

```
?  ?  ?  ?  ?  ?  [ESPACIO]
"  H  O  L  A  "  ?  ?  ?  ?
?  ?  [ESPACIO]  A  ?  2  ?
?  ?  ?  ?  ?  [ESPACIO]  A
?  ?  ?  ?  ?  ?  ?  ?  ?
```

Ya estás en condiciones de identificar una parte del programa en BASIC, pero todavía te queda un buen número de interrogaciones. El cuadrado representa un espacio.

A lo mejor esperabas encontrar la palabra PRINT precediendo a «HOLA», pero en su lugar te has encontrado con el número 153. El ordenador utiliza los valores superiores a 127 para designar las **palabras clave** del BASIC. Si consultas la lista de estos valores en el manual de tu aparato, encontrarás que 153 es el valor (el llamado *token*) correspondiente a la instrucción PRINT. Ahora ya puedes identificar las siguientes **palabras clave**:

```
?  ?  ?  ?  ?  PRINT[ESPACIO]
"  H  O  L  A  "  ?  ?  ?  ?
?  LET  [ESPACIO]  A  =  2
?  ?  ?  ?  ?  PRINT[ESPACIO]
A  ?  ?  ?  ?  ?  STOP  ?  ?
?
```

Si ahora observas los bytes 3º, 17º, 27º y 35º, descubrirás la secuencia 10, 20, 30, 40. Estos bytes se utilizan para contener el byte inferior de los números de línea mientras que los bytes siguientes contienen los correspondientes bytes altos. Los números comprendidos entre 0 y 65535 se representan aquí mediante combinaciones de 2 bytes (en formato: byte inferior + 256 * byte superior). Los dos bytes que preceden a los números de línea, con el byte bajo también en primer lugar, son los punteros de comienzo de la línea siguiente. Por ejemplo, los dos primeros bytes 15 y 8 están apuntando a 15 + 256*8, que es igual a 2063.

- LOCALIZACION DEL INICIO DEL BASIC
- TRADUCIENDO LOS NUMEROS
- CODIGOS ASCII Y TOKENS
- COMO SE ALMACENA EL PROGRAMA

Como la dirección del primer byte es 2048, te resultará fácil echar la cuenta para comprobar que efectivamente la dirección 2063 es la que corresponde al principio del siguiente conjunto de punteros. Esto se muestra más claramente en el siguiente ejemplo. Observa que el ordenador utiliza un 0 para marcar el final de línea (fdl):

```
0  ptrB  ptrA  10  0  PRINT
[ESPACIO]  "  H  O  L  A  "
fdl  ptrB  ptrA  20  0  LET
[ESPACIO] A  =  2  fdl  ptrB
ptrA  30  0  PRINT [ESPACIO]
A  fdl  ptrB  ptrA  40  0
STOP  fdl  0  0
```

siendo ptrI=byte inferior del puntero y ptrS=byte superior del puntero.

El ordenador señala el final del programa haciendo igual a cero los campos del puntero (ptrI y ptrS), por lo que un programa en BASIC siempre termina con tres ceros.

Cuando tecleas el comando NEW, los dos primeros bytes del puntero se ponen a cero, por lo que es muy fácil resetearlos. El puntero alto siempre es 8, mientras que el puntero bajo debe ser igual a la longitud de la primera línea, contando desde cero hasta la marca de final de línea.

ALMACENAMIENTO DE LAS VARIABLES

El **Commodore** dispone de tres zonas en las que puede almacenar variables. Las variables sencillas se almacenan a continuación del programa en BASIC, y las direcciones 45 y 46 contienen las direcciones del principio de la zona de almacenamiento. A continuación se almacenan las variables de tipo matriz, con la dirección de comienzo contenida en las direcciones 47 y 48. Hay otras dos direcciones —49 y 50— que apuntan al final de la zona de almacenamiento de variables. Las cadenas de caracteres se almacenan en el otro extremo de la RAM. La dirección de comienzo está dada por el contenido de las direcciones 51 y 52, mientras que las direcciones finales se almacenan en 53 y 54.

VARIABLES SIMPLES

Todas estas direcciones te permiten encontrar dónde se hallan exactamente las variables. El siguiente programa explora toda la zona de almacenamiento de variables simples para ver qué le sucede a la variable A. Los números que figuran en la línea inferior indican el contenido de la zona de almacenamiento, así como el código ASCII cuya traducción te incluimos en los casos en que se presenta:

```
1 AB=123
10 FOR Z=PEEK(45)+PEEK(46)*
256 TO PEEK(47)+PEEK(48)*256
-1
20 PRINT PEEK(Z);:NEXT Z

65  66  135  118  0  0  0
 A   B  ......valor......
```

Los siete últimos bytes almacenan la variable Z y constituyen la parte de programa utilizada para observar la memoria. Para evitar confusiones, omitiremos estos bytes de ahora en adelante. La variable en la que estamos interesados es AB, que se almacena simplemente con los códigos ASCII de A y B, seguidos del valor de la variable en formato de punto (coma) flotante.

Si en la línea 1 cambias ahora a una variable de cadena de caracteres, obtendrás los siguientes números:

```
90  0  140  5  80  0  0
 Z  .......valor.......
```

En primer lugar figuran los códigos ASCII del nombre de variable ZZ, pero al segundo número se le ha sumado 128 para indicar que se trata de una variable de cadena de caracteres. El siguiente número da el número de caracteres que contiene dicha cadena y los dos bytes siguientes son los punteros bajo y alto que apuntan a la posición de la cadena en la memoria. En este caso el valor es 10 + 8*256, que es igual a 2058. Esta es de hecho la posición de la palabra TEXT en el programa BASIC. Sin embargo, si dicha variable cambiara a lo largo del programa, o si estuviera inicialmente sin definir (como ocurre en una sentencia INPUT), los punteros estarían apuntando al lugar de almacenamiento de cadenas que está en el extremo superior de la RAM.

VARIABLES TIPO MATRIZ

Una matriz de cadenas de caracteres se almacena de forma parecida a una cadena. Ensaya el siguiente programa y lo verás. Fíjate en que los punteros de la línea 10 están ahora apuntando al principio y al final del almacenamiento de matrices.

```
1 ZZ$="TEXTO"

90  218  4  10  8  0  0  90
 Z    Z

218  5  10  8  0  0  90  0

140  5  160  0  0
```

Como de costumbre, comienza con el nombre, y también esta vez se le han sumado 128 a la segunda letra por tratarse de una variable de cadena de caracteres. El siguiente número, 16, es la cantidad de bytes que ocupa el almacenamiento. Cuéntalos y comprobarás que, efectivamente, son 16. Los dos bytes siguientes nos dan el número de dimensiones y el siguiente par el número de elementos reservados por la sentencia DIM. A continuación de estos números hay tres bytes por elemento. Como el primer elemento AA$(0) no está definido, los tres primeros bytes son cero. En cuanto a los otros, el primer byte da el número de caracteres de la cadena y los otros dos apuntan a la posición de la misma.

Por último, haz una prueba con una matriz numérica, en este caso una de dos dimensiones:

```
1 DIM AA$(2)
2 AA$(1)="MANZANA"
3 AA$(2)="CIRUELA"
10 FOR Z=PEEK(47)+PEEK(48)*
256 TO PEEK(49)+PEEK(50)*256
-1
20 PRINT#3,PEEK(Z);:NEXT Z

65  193  16  0  1  0  3  0

0  0  7  26  8  7  47  8
```

Ahora ya debes estar en condiciones de poder descifrar por tí mismo los primeros cinco bytes. Los dos pares de bytes siguientes contienen el número de elementos de cada dimensión, pero en orden inverso al de la sentencia DIM. A continuación figuran los valores de las variables en formato de punto flotante.

```
1 DIM AA(0,2)
2 AA(0,0)=1
3 AA(0,2)=2
10 FOR Z=PEEK(47)+PEEK(48)*
256 TO PEEK(49)+PEEK(50)*256
-1
20 PRINT PEEK(Z);:NEXT Z

65  65  24  0  2  0  3  0  1

129  0  0  0  0
.....valor.....

0  0  0  0  0
....valor...

130  0  0  0  0
.....valor.....
```

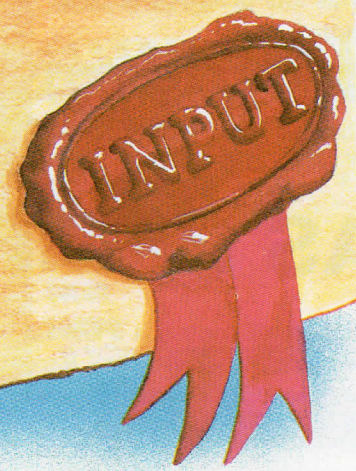
Si se te hace difícil encontrar INPUT
en tu kiosco habitual,
resérvalo por adelantado, o háznoslo saber
para que podamos remediarlo

LOS MEJORES DE INPUT COMMODORE

PUESTO	TITULO	PORCENTAJE
1.º	Hardball	19,3 %
2.º	Commando	15,1 %
3.º	Rambo	13,6 %
4.º	Desert Fox	12,3 %
5.º	Saboteur	10,9 %
6.º	Time tunnel	8,2 %
7.º	Kung Fu master	5,6 %
8.º	Skyfox	5,5 %
9.º	Green Beret	5,4 %
10.º	Misión Imposible	4,1 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Julio de 1986



BALL BLAZER: FUTBOL COSMICO

Alguien anuncia: «¡Atención señoras y señores! Dentro de una milesima de *Parsec* dará comienzo el deporte del siglo 31: **Ballblazer**, el juego con reminiscencias de las largas guerras espaciales de los siglos anteriores». Nos hallamos en el año 3097 en el sistema de soles gemelos **Kalaxon** y **Kalamar** y el deporte supranacional es ahora el **Ballblazer**, heredero directo del fútbol, que puede jugarse en este lugar privilegiado que carece de gravedad. El juego utiliza dos vehículos para desplazarse por una rejilla tras una pelota de grandes dimensiones.

Para los aficionados un poco despistados, repasaremos las reglas del juego. Dos participantes se enfrentan en una pista cuadriculada de 275 metros por 105. Cada cuadrado tiene un tamaño de 5 por 5 metros. La pista es curva, de tal modo que un objeto que se halle a dos metros de altura dejará de verse a partir de una distancia de 17 cuadrados. Existe una barrera eléctrica que rodea al campo, evitando que los dos contrincantes abandonen el juego.

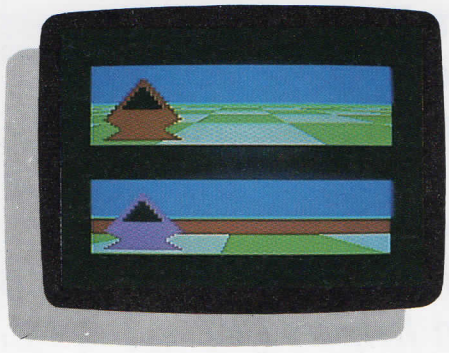
El fin del juego es tan simple como introducir una pelota magnética de 8 metros de diámetro y una tonelada de peso en una portería móvil. Una de las dificultades proviene del hecho de que la pelota se desplaza a una velocidad de crucero de 600 metros por segundo.

Otra dificultad es que la separación entre poste y poste de la portería aumenta conforme se marcan los goles. La distancia inicial de separación es de 12,5 metros.

Ambos contrincantes utilizan vehículos supersofisticados, llamados **Rotofoils**, de dos metros y medio de diámetro y capaces de alcanzar una velocidad de 50 metros por segundo. Su peso es de tres toneladas. El **Rotofoil** dispone de un ordenador a bordo que calcula la distancia que separa a la pelota y la nave, atrayéndola por medio de un sofisticado sistema de unión por campo magnético.

La pantalla está dividida en dos partes, para que cada jugador vea las acciones del contrario, por lo que es importante poder vigilar ambas mitades. Hay que tener cuidado, pues el oponente tratará de quitarte la bola igual que tú lo hiciste con él.

Los gráficos ofrecen una visión tridimensional realista y pueden elogiarse como buenos. Por otro lado, este juego goza de la característica de la adictividad.



DATOS GENERALES

TITULO Ballblazer

FABRICANTE Lucasfilm/Activision

ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Juego de ciencia ficción

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	8,5
GRAFICOS	8,5
COLOR	8,5
SONIDO	8,5
TOTAL	43

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

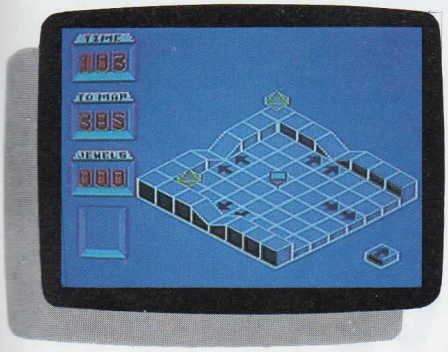
LAS RAMPAS TRAIDORAS

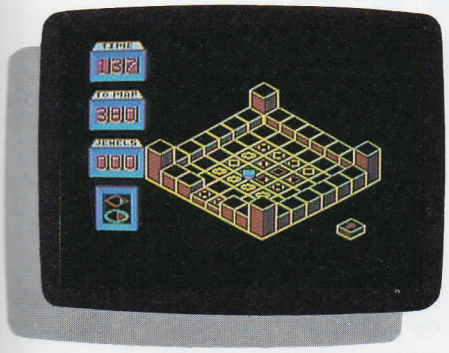
En **Spindizzy** tú controlas a **Gerald**, (*Geographical Environmental Recce and Landmapping Device*) que puede tomar la forma de una pirámide invertida, una pelota o algo similar a un giroscopio.

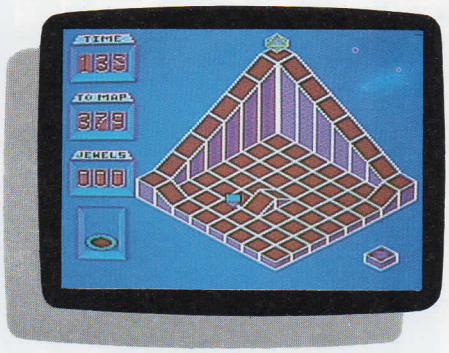
El juego pretende que demuestres tu habilidad en el avance a través de pantallas con paisajes geométricos tridimensionales.

Existen 389 pantallas, por las que se puede desplazar **Gerald**. En su

movimiento giratorio debe recoger todos los cristales esparcidos por doquier. El constante avance del







reloj y el pequeño margen de error que se te permite son tus dos principales enemigos. Cada vez que recoges un cristal se te abonan diez segundos, pero si te caes por un lateral perderás la preciada energía. Con tan gran número de pantallas, se puede asumir que nos movemos por un laberinto, en el que debemos evitar las peligrosas caídas. Hay gran cantidad de rampas, conmutadores y ascensores. En algunos lugares, tu punto de vista te impedirá ver lo que estás haciendo. Para utilizar un ascensor hay que haber activado previamente el conmutador, llegando hasta él. Pero cuidado, algunos también alteran nuestro entorno y cambian el aspecto de la pantalla. Un consejo, es casi imprescindible ir elaborando el mapa según avanzas, si no quieres verte perdido.
El efecto tridimensional es fantástico, así como la suavidad del movimiento. Sin embargo el sonido no es uno de los puntos fuertes de este entretenido programa.

DATOS GENERALES

TITULO Spindizzy

FABRICANTE Electric Dreams

ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Laberinto Tridimensional con rampas

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8,5
INTERES	9
GRAFICOS	8,5
COLOR	8,5
SONIDO	7
TOTAL	41

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

NEXUS CONECTION

Nexus es un juego en el que aparece el *kung fu* como telón de fondo.
Eres un periodísta con olfato que intentas infiltrarte en el cuartel central del tráfico de drogas, buscando a un amigo desaparecido y pruebas para escribir un buen artículo. Estás convencido de que tu amigo se encuentra en aquellas instalaciones.
El *joystick* basta para lanzar patadas y puñetazos, correr, dar volteretas... así como para desplazarte por los pasadizos subterráneos de la malvada organización.
Tienes contactos internos que te irán enseñando cuánto saben, utilizar nuevas armas o patadas más complejas.
Debes ir recolectando las 128 pistas informativas, dispersas por todas partes y envíarle el resultado a tu director. Habrás de llegar a la sala de transmisiones y enviar por un terminal tus respuestas al **Daily Clarion**.
En este juego aparecen fotos digitalizadas del personaje que se encuentra en la pantalla, algo similar a lo que ocurre con **Regreso al futuro**.
Los gráficos son de calidad impresionante y la pantalla se divide en áreas que muestran el rostro de los personajes, donde se hallan, lo que hacen y hasta los distintos mensajes que necesitas para progresar en el juego.
El sonido también ha sido elaborado a conciencia.

DATOS GENERALES

TITULO Nexus

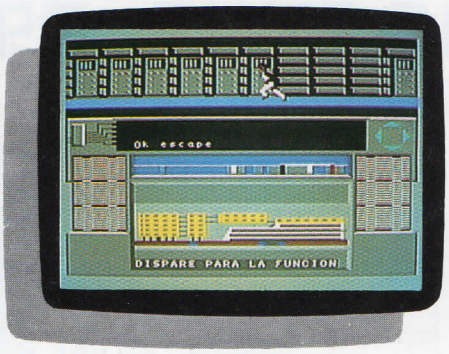
FABRICANTE Nexus Production Ltd.

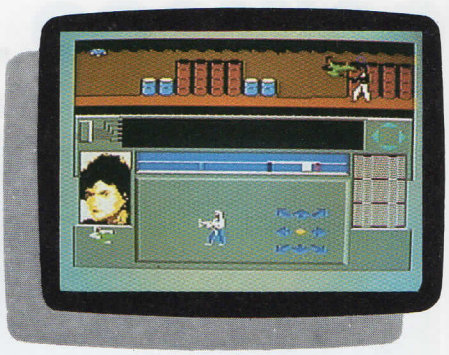
ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Acción Antidroga

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	9
GRAFICOS	9,5
COLOR	8,5
SONIDO	8
TOTAL	44





¿DONDE ESTA LA LLAVE DE LA MOTO?

DATOS GENERALES

TITULO Kick Star

FABRICANTE Mastertronics

ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Pruebas de habilidad con moto

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**7,5**
INTERES	**8,5**
GRAFICOS	**9**
COLOR	**8,5**
SONIDO	**7,5**
TOTAL	**41**

Arranque rápido es un juego de habilidad. Manejando con destreza el manillar de una moto todoterreno, debes salvar los obstáculos que encuentras a lo largo de dieciséis pantallas. La destreza es el factor decisivo.

La moto se desplaza hacia la derecha de la pantalla y el terreno se hace más difícil a medida que avanzas. Las pruebas incluyen saltos, terreno pedregoso, avances entre llantas, arbustos y barriles, paredes de ladrillo, muros y puertas. Hay que tener muchisimo cuidado para no caer. Asi que deberás entrenarte unas cuantas veces, hasta tomrle el pulso al juego.

Dispones de un total de ocho pistas de entrenamiento y has de organizar carreras combinando tres de ellas.

El programa te permite hacer cosas tales como el caballito. Las caídas que puedes sufrir si cometes un error son espectaculares. Por otro lado, puedes estropear un total de cinco motos. También puedes competir en habilidad con otro piloto.

Hay dos elementos poco agradables. El primero es que en caso de caída el programa desplaza la moto a la parte inferior de la pantalla; zona en la que ritmo, velocidad y manejo de la moto son mínimos. Esto se mantiene hasta el principio de otra serie de obstáculos; a veces hay que esperar un buen rato. Lo segundo es que en caso de elegir un sólo jugador éste compite contra sí mismo. El programa no incluye competición contra el ordenador. Los gráficos son del tipo habitual en la serie **Mastertronic**.

La pantalla aparece dividida en dos segmentos, uno para cada jugador.

Las sorpresas que van surgiendo hacen que sea un juego entretenido y adictivo.

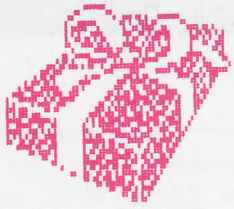
CONOCE A FONDO TU COMMODORE

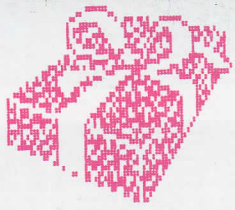
Aprende fácilmente en un curso de verano a
- Programar en Basic
- Hacer tus propios Sprites y Juegos
- Hacer tus propios Programas musicales
- Trabajar con ficheros Secuenciales y Relativos
- DANDO CLASES CON COMMODORE 64 ó 128
- CON UN ORDENADOR POR ALUMNO
- EN GRUPOS REDUCIDOS

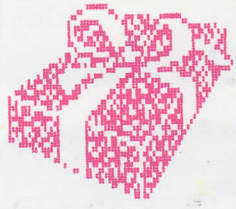
en INPUT SA
C/ Armonia, 22 Teléfono: 7410987
APROVECHA LA OPORTUNIDAD. LLAMANOS

Reserva tu INPUT

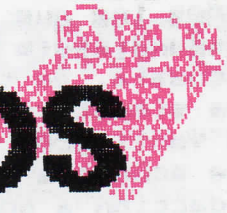
ESPECIAL de Verano







Sortearemos mas de 100 REGALOS SORPRESA



ENTREGA DE UNO DE LOS PREMIOS DEL CONCURSO DE ARTE POR ORDENADOR

En la foto adjunta podemos ver el momento en el que Sebastián Vila, ganador de uno de los segundos premios del concurso de Arte por ordenador, recoge durante el pasado Informat sus regalos. La entrega fue realizada por Máximo Cabezas, de la firma Microbyte. Enhorabuena.

LA AFORTUNADA GANADORA DEL COMMODORE 64 RECOGE SU ORDENADOR

En el pasado número 9 de INPUT Commodore publicabamos el nombre del ganador del sorteo de un Commodore 64 entre todos los que respondisteis a la encuesta. Se trata de una simpática mujer de ventipocos años, que acudió a nuestra redacción a por su premio.
En la foto, Jacinto Tosca hace la entrega pertinente.

EN BUSCA DEL TITANIC

Con este juego puedes revivir las aventuras de quienes buscaron afanosamente los restos sumergidos del majestuoso **Titanic**.
El objetivo está claro, reflotar los restos del barco y localizar la caja fuerte que se supone alberga una fortuna en dinero y joyas. Para llevar a cabo la parte delicada del trabajo, dispones de un sofisticado dispositivo autónomo de navegación submarina, que dispone de un brazo robótico y cámaras de televisión. Lo puedes controlar voluntariamente a medida que recibes la información desde el fondo del océano.
Como siempre, hay un lado político en el asunto y debes pelear contra el tiempo para obener los resultados que pueda airear la prensa, para que tus patrocinadores continúen poniendo dinero. Por otro lado, sólamente puedes sumergir el robot una vez al día.
El brazo tiene otro tipo de limitaciones, tales como el número de objetos que puede recoger, un total de cuatro por vez.
La pantalla se divide en áreas. En la parte central aparece un enorme calendario, que marca el avance de los días. Distintas ventanas tienen cometidos específicos. La financiera te pone en contacto con tus patrocinadores. Otra te permite dar ruedas de prensa. En la ventana lunar puedes jugar con el avance del tiempo, haciendo que las fases evolucionen con mayor rapidez.
Consigues finalizar el juego cuando inflas los ocho enormes flotadores que sacan el barco a la superficie.
Este juego, aún cuando es de estrategia, dista mucho de ser aburrido. Todo lo contrario, has de estar permanentemente concentrado.

DATOS GENERALES

TITULO Titanic

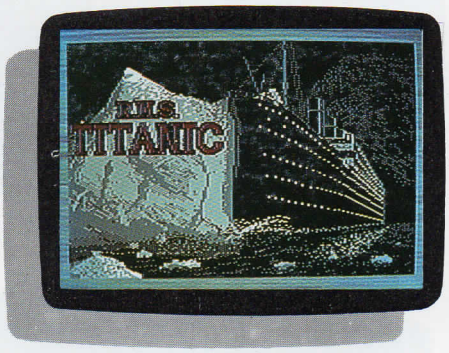
FABRICANTE Electric Dreams

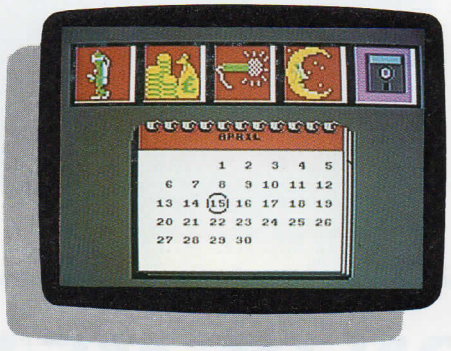
ORDENADOR Commodore 64

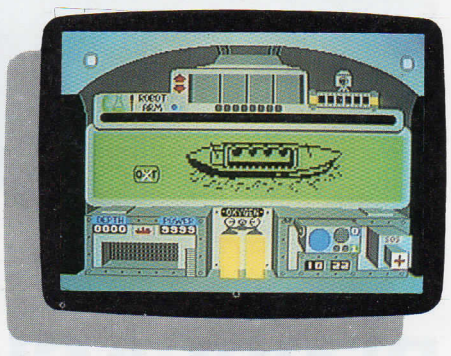
TEMA DEL PROGRAMA
Rescate Submarino

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**9**
INTERES	**9**
GRAFICOS	**8**
COLOR	**7,5**
SONIDO	**7,5**
TOTAL	**41**







★★★★★★★★★★★★★★★★★★★★★★★★★★★★

NAVE ESTRELLA ANDROMEDA

El juego se compone de cinco secciones. Deben ser cargadas y jugar con ellas por separado. Cada una incluye un paseo por el metaespacio y la búsqueda de una clave de acceso que permite pasar a la siguiente sección.
Estamos ante un elaborado juego que recuerda algo al popular y complejo **Elite**. En él vagas por el espacio en busca de estaciones habitadas, tratando de llegar hasta el lugar en que se encuentran. En muchos lugares no cae bien tu entrada.
Alana es una dictadora que gobierna la **Federación Galáctica**. Dispone de una llave de seguridad, que no es más que un cristal de composición desconocida que permite el acceso al todopoderoso ordenador **Mindlord** (Señor de la Mente), y dominador de la galaxia.
Debes buscar una lanza de protones y los dos elementos que la activan, el **Zyron** y el **Trysst**.
Ayudados por la lanza, se puede perseguir a **Alana** y desposeerla del cristal colgado de su cuello.
No sabes hacia donde ir, pero debes estar pendiente del *scanner* de comunicaciones, a la espera de conocer un mensaje que te dará una dirección (los mensajes que emplea **Starship** han sido minuciosamente traducidos a nuestra lengua, para no

DATOS GENERALES

TITULO Starship Andromeda

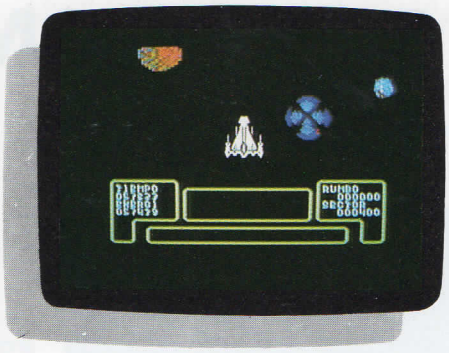
FABRICANTE Ariolasoft

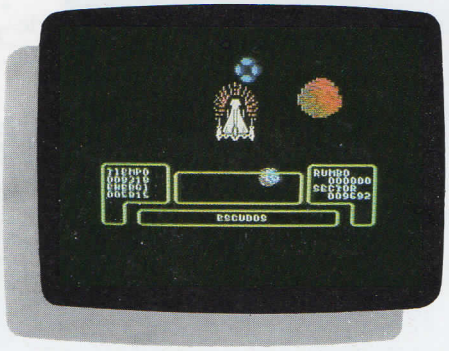
ORDENADOR Commodore 64

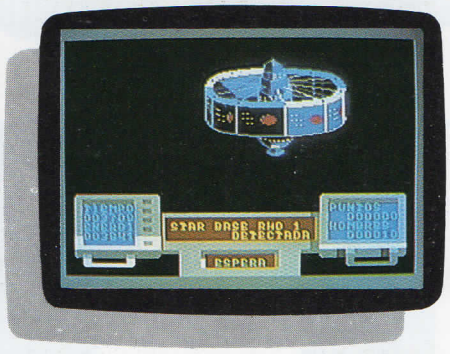
TEMA DEL PROGRAMA
Pruebas de habilidad en el espacio

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**9**
INTERES	**9**
GRAFICOS	**8,8**
COLOR	**8**
SONIDO	**8,5**
TOTAL	**43,3**







restarle comprensión al programa). Conseguido esto, sales del espacio real para entrar en el metaespacio, donde puedes recorrer grandes distancias en breves intervalos de tiempo. Has de ser cuidadoso para no estrellarte contra los meteoritos y asteroides. Aunque dispones de un escudo protector que te librará de ellos cuando lo actives.
El combate con las naves enemigas no es difícil, simplemente hay que permanecer en un sitio y esperar a que vayan pasando para destruirlas.
Llegar al final de este juego no es tarea fácil. La energía debe dosificarse cuidadosamente.
En la primera parte tienes que atravesar túneles estrechos, utilizando los propulsores de la nave para no rozar las paredes. Luego encuentras tres naves gobernadas por androides, debes ir de una a otra.
Después tienes que enterarte de una serie de códigos de viaje, explorando las comunicaciones por radio de la nave estrella.
A continuación tienes que recomponer una especie de rompecabezas. Necesitarás mucho tiempo para llegar al final.
Los conseguidos efectos sonoros y la música son un acicate perfecto.
El movimiento de naves y meteoritos está muy bien resuelto.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

RECREACION DE MEJICO 86

Dicen que lo que ocurrió una vez no puede volver a repetirse. Sin embargo esto no es siempre absolutamente cierto. Por ejemplo, podemos hacer que los resultados del **Mundial** de fútbol puedan ser diferentes a como fueron, simplemente repitiendo los partidos.

World Cup Carnival es una recreación de todo lo jugado en Méjico. Están presentes las selecciones que fueron capaces de llegar hasta allí.
Una vez cargado el programa, participan dos jugadores simultáneamente que pueden entrenar o lanzarse al juego. No obstante, al principio se definirá cuál es el número total de jugadores, hasta ocho, que elegirán uno de los diez equipos participantes.
Este programa sigue la misma línea del popular **Soccer** que popularizó **Commodore**. Sin embargo ofrece mejoras con respecto a aquel. Los *sprites* de los jugadores están más elaborados y los uniformes responden a las selecciones nacionales de los paises participantes.
Durante el entrenamiento podemos dedicarnos al lanzamiento de *penalties* o aprender a controlar el Salón.
Están previstos muchos golpes, desde el pie o la rodilla hasta los de

DATOS GENERALES

TITULO World Cup Carnival

FABRICANTE U.S. Gold

ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Mundiales de Méjico 86

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**8,5**
INTERES	**9**
GRAFICOS	**9,5**
COLOR	**8,5**
SONIDO	**8**
TOTAL	**43,5**

El disparo se produce con el botón del *joystick*. Acabado cada partido, aparece la clasificación provisional del torneo.
U.S. Gold ha negociado con la **FIFA** la inclusión del trofeo y demás emblemas en el juego y toda la parafernalia que acompaña al *cassette*, desde un *poster* con las efemérides de los Mundiales, hasta un escudo de tela. También se

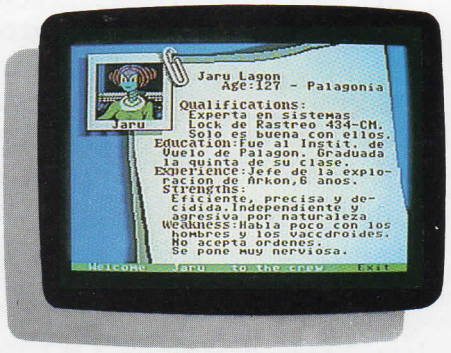
cabeza. La pelota será controlada o perseguida por el jugador de cada equipo que se encuentre más próximo al balón en cada momento.

incluye un cartel utilizable para dar seguimiento diario a los partidos y un conjunto de banderas adhesivas de los paises, que se pegarán a medida que se aproxima la final.
En definitiva **World Cup** es una acertada actualización de un juego clásico en el **Commodore 64**.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

COMPAÑIA COMERCIAL PSI5

Se hizo esperar, pero ha llegado. Este juego ha creado gran expectación en los últimos tiempos, después de haber circulado un *prerelease* que perfilaba las características del programa.
Por fin eres lo que soñabas, capitán de una nave de carga en la compañía de comercio **Psi**. Sólo te falta la tripulación y debes elegir cinco entre treinta aspirantes. Habrás de estudiar sus *curriculums*, en los que se adjunta la obligada foto, y determinar cuál es el equipo de mayor confianza.
Los candidatos aparecen agrupados en páginas de seis fotos. Eliges con el *joystick* a los que van interesándote y puedes ver su expediente mecanografiado, en el que se incluyen sus cualificaciones, formación, experiencia, puntos fuertes y débiles. Así que eres juez único de sus aptitudes para interpretar tus órdenes. Las equivocaciones se pagan.
Este no es un simple juego, hay que ser rápido de reflejos y con mente ágil para llegar a buen puerto. Debes aprender a interpretar los menús y mensajes sin dilación. La facilidad con que seas capaz de desplazarte entre los menús es clave para no recibir sorpresas. La pantalla se divide en cuatro áreas fundamentales. Las dos superiores son sendas pantallas, por las que puedes ver el entorno exterior de la nave, así como a quien te ataca y establecer contacto visual con los tripulantes en su puesto de trabajo. Se trata de un videoteléfono en el



que podemos ver como el interlocutor mueve sus labios para hablar.
La parte central está ocupada por el indicador de mensajes, la brújula, el velocímetro y el indicador de la efectividad de las armas.

DATOS GENERALES

TITULO Psi 5 Compañía Comercial

FABRICANTE Accolade

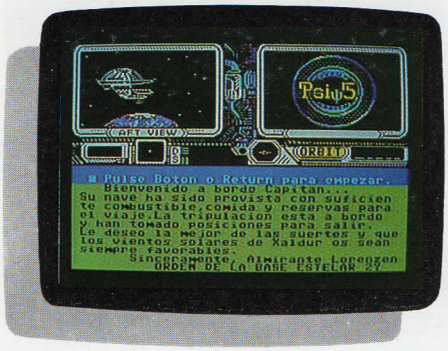
ORDENADOR Commodore 64

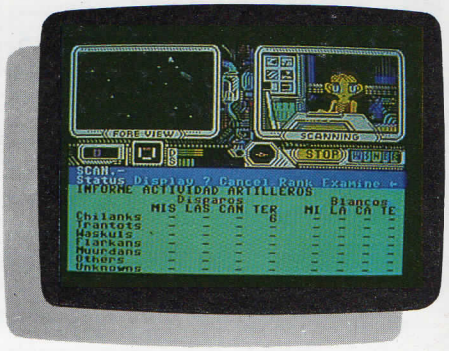
TEMA DEL PROGRAMA
Manejo de nave sofisticada

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**9,5**
INTERES	**9**
GRAFICOS	**9**
COLOR	**8,5**
SONIDO	**8,5**
TOTAL	**44,5**

La última zona ocupa casi la mitad inferior. En ella aparece el menú de opciones en una banda, en el que puedes elegir lo que hacer en cada momento, y los continuos informes para el gobierno de la nave. Decides lo que hacer en cada momento en base a las circusnstacias, avanzando entre los menús y submenús. Los informes contribuyen a que tengas una ayuda en la toma de decisiones. Por ejemplo, el *scanner* te sugerirá cuál es el tipo idóneo de arma a utilizar contra un determinado enemigo.
Hay que afrontar tres misiones, cada una de ellas incremnta el riesgo, la distancia y el premio. El manejo fundamental se realiza fácilmente con el *joystick*.
Si te equivocaste de tripulación, siempre puedes volver a encender el ordenador y cargar el programa. Seguro que al final darás con una de confianza.
Psi 5 es un juego de estrategia y razonamiento, mezclado con elementos de *arcadia*. La necesidad de razonar continuamente lo hace altamente entretenido. Su forma de obligar a que el usuario participe lo hace parecer una novela gráfica interactiva. Una nota final. Dado que el manejo de textos es una parte importante del juego, la versión recién lanzada está adaptada al castellano en lo que se refiere al manejo.





EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.
b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

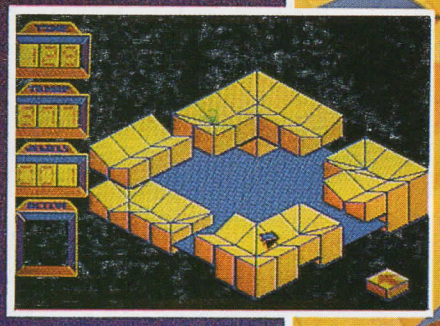
Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

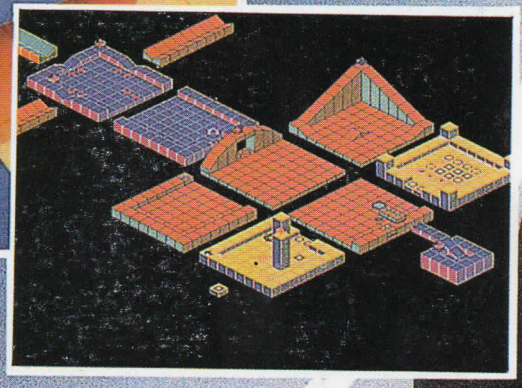
Envía tu mensaje a:

INPUT COMMODORE-ZOCO
c./ Alberto Alcocer, 46
28016 **MADRID**

LO MEJOR PARA COMMODORE

Spindizzy





EL RETO A TU DESTREZA EN TRES DIMENSIONES

— Con SPINDIZZY te trasportaras a un enigmático mundo de fantasía.
— Sus casi 400 pantallas se convierten en fuente inagotable de diversión.
— Los mejores gráficos tridimensionales que un AMSTRAD puede generar en un juego de este tipo.
— Tu mismo puedes alterar el trazado de las pantallas, situándote cada vez ante un nuevo desafío: rampas, ascensores, trampas, etc.

TAMBIEN DISPONIBLE PARA AMSTRAD Y SPECTRUM

Electric Dreams SOFTWARE

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES, O DIRECTAMENTE POR CORREO O TELEFONO A:

PROEIN, S.A.

Velázquez, 10 · 28001 Madrid Tels. (91) 276 22 08/09

INPUT commodore

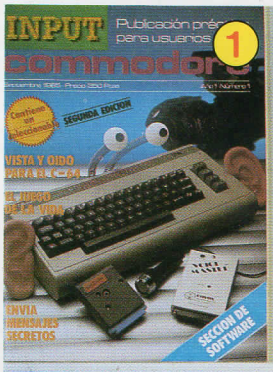
SERVICIO DE EJEMPLARES ATRASADOS

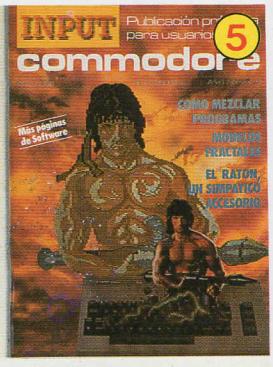
¡NO TE PIERDAS NI UN SOLO EJEMPLAR!

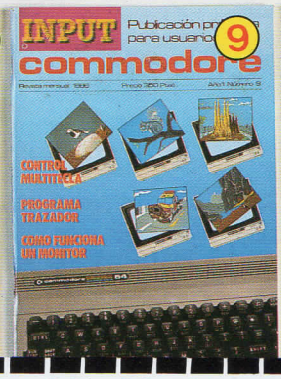
INPUT COMMODORE quiere proporcionar a sus lectores este nuevo servicio de ejemplares atrasados para que no pierdan la oportunidad de tener en sus hogares todos los ejemplares de esta revista, líder en el mercado español.

Podréis solicitar cualquier número de **INPUT COMMODORE** que querais, siempre al precio de cubierta (sin más gastos).

Utiliza el cupón adjunto, enviándolo a **EDISA** (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid, o bien llámanos por teléfono al (91) 415 97 12.







INPUT commodore

CUPON DE PEDIDO

SI, envíenme contrarreembolso ..... ejemplares de **INPUT COMMODORE** de los números:

(marca con una (X) tu elección)

1	2	3	4	5	6	7	8	9	10

NOMBRE

APELLIDOS

DOMICILIO

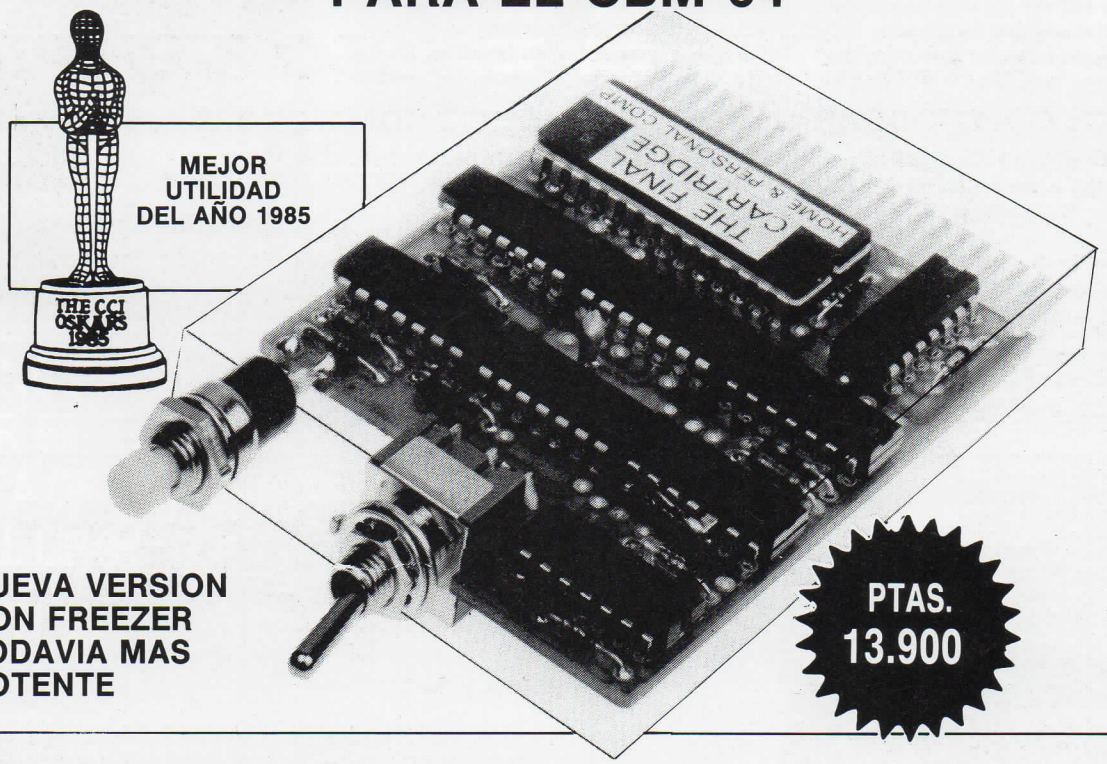
NUM. PISO ESCALERA COD. POSTAL

POBLACION PROV.

TELEFONO FIRMA

THE FINAL CARTRIDGE ®

EL PRIMER SISTEMA OPERATIVO EXTERNO PARA EL CBM 64



NUEVA VERSION CON FREEZER TODAVIA MAS POTENTE

PTAS. 13.900

Este nuevo sistema operativo construido en un cartucho no utiliza memoria alguna y está siempre presente. Compatible con 98% de los programas.

TURBO DISCO: Carga 6 veces más rápido - Salva 8 veces más rápido - No Borra la pantalla.

TURBO CINTA: 10 veces más rápido, incluso con ficheros. Utiliza los comandos normales del Commodore (Load-Save-Input#,...).

INTERFACE CENTRONICS: Compatible con las impresoras de tipo paralelo. Imprime los gráficos Commodore y los códigos de control. (Importante para los listados).

VOLCADOS DE PANTALLA: De alta resolución y texto. 1 página de ancho. 12 tonos de grises, copia pantallas de juegos o de programas como Doodle, Koala Pad, Print Shop, etc. Busca automáticamente la dirección en memoria.
Versión especial para impresoras MPS 801 y 803.

24 K. MAS DESDE EL BASIC: 2 nuevos comandos, "memory write" y "memory read" mueven 192 bytes muy rápidamente en cualquier sitio de los 64K RAM del CBM 64. Se pueden usar con cadenas y variables.

COMANDOS DEL BASIC 4.0: como dload, dsave, dappend, catalog, etc.

TECLAS DE FUNCION PREPROGRAMADAS: Run, load, save, catalog, comandos de disco, list (quita las protecciones contra el listado).

Ejemplo de volcado de pantalla.

OTRAS AYUDAS: Puede borrar parte de una línea, parar y continuar los listados, mover el cursor a la izquierda.

MONITOR DE CODIGO MAQUINA: Scroll hacia arriba y abajo. Bankswitching (para leer y escribir debajo de las ROMS), etc... No reside en memoria.

BOTON DE RESET: Resetea el monitor.

Resetea con volcados de pantalla. Resetea todos los programas protegidos.

INTERRUPTOR ON/OFF: Por si acaso...

FREEZER:

Para y continúa casi todos los programas. Permite hacer una copia del programa a disco o a cinta automáticamente.

—copia todo el programa en 1 bloque.
—es 4 ó 6 veces más rápido que los productos (similares) y especializados que se pueden encontrar en el mercado americano.

Funcionamiento sencillo por menús.
Permite: —impresión página entera.
—cambio de color letras o fondo.
—modo normal o inverso.
—salto a monitor o reset.
También, modo entrenamiento para juegos (anula la detección de colisión de los sprites).

Disponible en las mejores tiendas o directamente por correo a:

IMPORTADOR EXCLUSIVO

HISPASOFT, S.A.

C/ Coso 87-6.º A - Telf. (976) 39 99 61 - 50001 ZARAGOZA

copyright and registered trademark H&P computers
Wolphaertsbocht 236 3083 MV Rotterdam Netherlands. Tel.: 01031 - 10231982 Telex 26401 a intx nl

CONCURSO INPUT/ANAYA MULTIMEDIA

¡DESCUBRE AL ASESINO!

BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

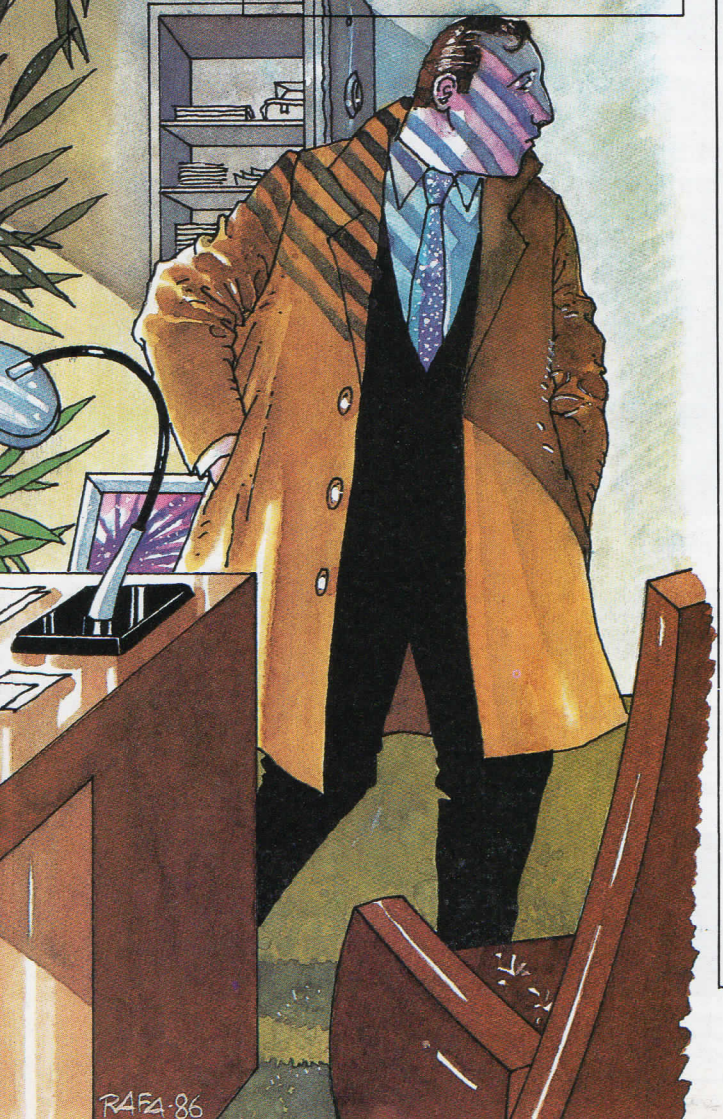
La solución del tercer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de agosto.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.

Quienes deseeis optar al premio especial debeis enviar en sobre aparte las respuestas a todos los mensajes. La fecha límite es también el 15 de agosto. El afortunado recibirá la noticia por teléfono, ya que el PCW Show *se celebra a principios de septiembre. Su nombre aparecerá publicado en INPUT de octubre.*

NOTA:

El plazo para la recepción del segundo mensaje es el próximo 15 de julio, aunque por error apareció la misma fecha que para el primero.



EL CASO DEL ANCIANO ASESINADO

—Me siento culpable —decía entre lágrima y lágrima—. Si yo hubiera estado con él, tal vez no habría muerto.

Intenté consolarle, pero la verdad es que el viejo me producía cierta repugnancia. Anoté:

CSQESOXDO MSPBKNY JSQ JKQ DWVOABKSELBKNV
CEKKRMODBOCCYYOZBCOXCONMKCBPXROBOYNBYVMB
ESKNVVMOYVVSKWCCKKKWSQKDC

Por lo demás, también vivía solo, como su hermano. Al parecer, los dos eran los últimos miembros de una familia de rancio abolengo, cuya grandeza había entrado en franco declive. De sus antiguas riquezas apenas quedaban ya unos restos: tres cuadros de firmas importantes, una pequeña casa junto al mar que casi no podían mantener y las joyas que habían robado de la caja fuerte de su hermano.

—Esto es el fin —dijo el pobre viejo entre sollozos—, esto es el fin de una importante dinastía y Dios ha permitido que viva para verlo.

Cuando llegué a la comisaría, mi ayudante había salido, pero sobre la mesa de mi despacho me había dejado una nota con el siguiente texto:

PFDRFBKQB ZFCOXAL QOXKPMLPFZFLK FQFKBOXO
FL BKIDOBOFLZBPBQAQCRLKBXBIXZXXIZPAXQXBS
IPXRXYBCAXIBXGPIALRLBIALJLWOFOWWQFWWWLWW
WWWWWWW

Junto a la nota estaba la colilla que, supuestamente, había originado el incendio y que, sin duda, era la clave del enigma. Una sonrisa se dibujó en mi rostro cuando anoté:

RHFTHDMSD BHEQZCN FQNMREDKC D WITGW DI G
VAES HFSXCQAI RAJMEFD LR MG HOPKOSE TK U
OHKDU HJYYIRIXPV QKWFIEWHQFTYE PCV TESIF
S HGMHHBY UOV NRZ MOINSMXRZ HF AS FYODKS
S TJRZKRZS

... Había resuelto un nuevo caso.

Juan José Millas

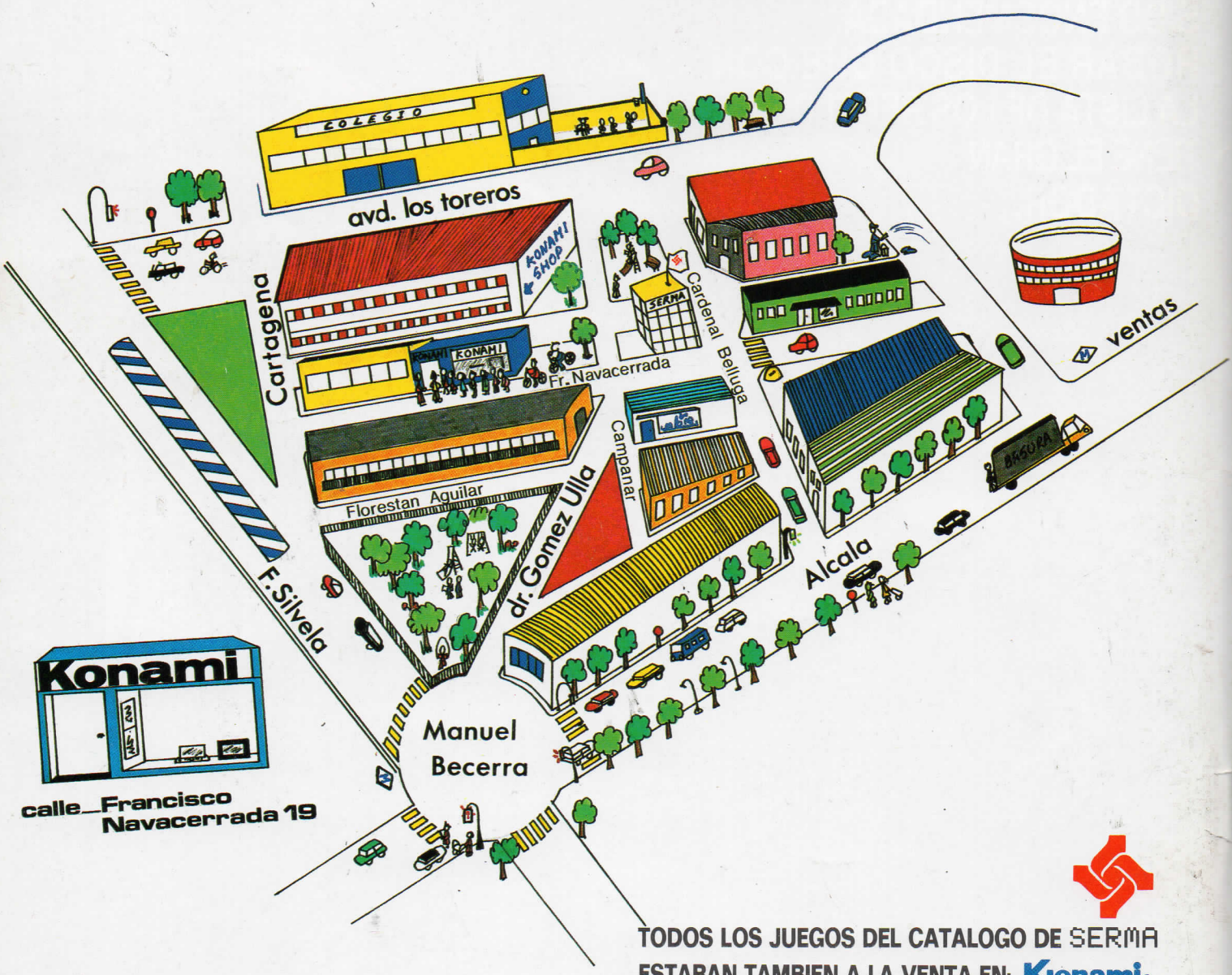
SI BUSCAS LO MEJOR ERBE Software LO TIENE

SABOTEUR!

COMO EXPERIMENTADO MERCENARIO CUIDADOSAMENTE ENTRENADO EN LAS ARTES MARCIALES, DEBES CUMPLIR LA MISION QUE TE HA SIDO ENCOMENDADA: ROBAR EL DISCO QUE CON LA LISTA DE LOS REBELDES TIENE EL GRAN DICTADOR.

Konami SHOP®

La marca nº1 en videojuegos para ordenador abre su primera tienda de Europa en Madrid



TODOS LOS JUEGOS DEL CATALOGO DE SERMA ESTARAN TAMBIEN A LA VENTA EN: Konami®

LOS TITULOS EDITADOS POR Konami® PARA SPECTRUM, CM 64, AMSTRAD Y MSX SON: FABRICADOS Y DISTRIBUIDOS EN EXCLUSIVA POR SERMA.

Rogamos a todos los usuarios de Spectrum, CM 64, Amstrad y MSX que nos escriban a SERMA sugiriendo ideas para hacerlas realidad en nuestra tienda